

PLATAFORMA WEB PARA LA ANOTACIÓN DE VÍDEO

TRABAJO FIN DE MÁSTER

Jaime Alba ([jaime.alba.cepero@usal.es](mailto:jaime.alba.cepero@usal.es))

MÁSTER EN INGENIERÍA INFORMÁTICA

UNIVERSIDAD DE SALAMANCA

13 de octubre de 2021

Contenido

[1 Introducción 5](#_Toc92802131)

[1.1 Agradecimientos 5](#_Toc92802132)

[2 Motivación 5](#_Toc92802133)

[2.1 Definición del tipo de trabajo 6](#_Toc92802134)

[2.2 Resumen de la propuesta inicial y comparación con el desarrollo final 7](#_Toc92802135)

[2.2.1 Anotaciones 7](#_Toc92802136)

[2.2.2 Formato de salida 8](#_Toc92802137)

[2.2.3 Versión de escritorio y versión móvil 8](#_Toc92802138)

[2.2.4 Características del entorno 8](#_Toc92802139)

[2.2.5 Enfoque de implementación 8](#_Toc92802140)

[2.2.6 Principales limitaciones 9](#_Toc92802141)

[2.3 La falsa seguridad del especialista 9](#_Toc92802142)

[2.4 Situación actual 10](#_Toc92802143)

[2.4.1 Generación de contenido de vídeo 10](#_Toc92802144)

[2.4.2 Productos de pago y licencias 10](#_Toc92802145)

[2.4.3 Uso particular / educativo. 10](#_Toc92802146)

[2.4.4 10](#_Toc92802147)

[3 Contexto 11](#_Toc92802148)

[3.1 Relación con objetivos y competencias del máster 11](#_Toc92802149)

[3.2 Relevancia social 12](#_Toc92802150)

[4 Antecedentes tecnológicos 12](#_Toc92802151)

[4.1 Evolución del concepto de edición 13](#_Toc92802152)

[4.2 Comparación con otros programas 15](#_Toc92802153)

[4.2.2 Elementos comunes de los editores 19](#_Toc92802154)

[4.3 Tendencias 19](#_Toc92802155)

[4.4 Nicho 19](#_Toc92802156)

[4.5 Objetivos y metodología 19](#_Toc92802157)

[5 Delimitación del objeto del trabajo 20](#_Toc92802158)

[5.1 Evolución del análisis de la usabilidad 20](#_Toc92802159)

[5.2 La usabilidad en el editor de vídeo 20](#_Toc92802160)

[5.3 Consideraciones para el éxito del producto 21](#_Toc92802161)

[5.3.1 Adaptación al usuario / cliente 21](#_Toc92802162)

[5.3.2 Modelo de negocio 26](#_Toc92802163)

[5.3.3 Competencia tecnológica 26](#_Toc92802164)

[5.3.4 Comentarios 26](#_Toc92802165)

[6 Diseño metodológico 27](#_Toc92802166)

[6.1 Búsqueda de necesidades 27](#_Toc92802167)

[6.1.1 Entrevistas 27](#_Toc92802168)

[6.1.2 Tiempo medio de uso de la aplicación 28](#_Toc92802169)

[6.1.3 Dispositivos de visualización 29](#_Toc92802170)

[6.1.4 Uso que se dará 29](#_Toc92802171)

[6.2 Diseño de personas 29](#_Toc92802172)

[6.3 Consideraciones de diseño 33](#_Toc92802173)

[6.3.1 242 Part II: Making Well-Behaved Products 33](#_Toc92802174)

[6.3.2 Impacto de primer uso y facilidad de descubrimiento 34](#_Toc92802175)

[6.3.3 Minimalismo 35](#_Toc92802176)

[6.3.4 Orquestación y disposición de elementos 35](#_Toc92802177)

[6.3.5 La barrera tecnológica y de nomenclatura 36](#_Toc92802178)

[6.3.6 Integración y diseño visual 42](#_Toc92802179)

[6.3.7 Complejidad de diseño vs. complejidad de uso 43](#_Toc92802180)

[6.4 Diseño software 44](#_Toc92802181)

[6.5 Pruebas de usabilidad 45](#_Toc92802182)

[6.6 Evaluación del producto 45](#_Toc92802183)

[6.6.1 Inspección (inspection) 45](#_Toc92802184)

[6.6.2 Indagación (assestment) 45](#_Toc92802185)

[6.6.3 Test 46](#_Toc92802186)

[6.6.4 Evaluación de rendimiento 48](#_Toc92802187)

[6.7 Consideraciones de compatibilidad 48](#_Toc92802188)

[6.8 Gestión de proyectos 51](#_Toc92802189)

[6.9 Línea de trabajo waterfall vs scrum 51](#_Toc92802190)

[7 Relación de dependencias 52](#_Toc92802191)

[7.1 Objetivo: emplear las menores dependencias posibles 52](#_Toc92802192)

[7.2 El caso FFMPEG 52](#_Toc92802193)

[7.3 Multiplataforma 52](#_Toc92802194)

[7.4 Requisitos hardware 52](#_Toc92802195)

[7.5 Requisitos software 52](#_Toc92802196)

[8 Tecnologías involucradas 53](#_Toc92802197)

[8.1 Entorno 53](#_Toc92802198)

[8.2 Cliente 54](#_Toc92802199)

[8.3 Servidor 54](#_Toc92802200)

[8.3.1 Máquinas de desarrollo y despliegue 55](#_Toc92802201)

[8.4 Interacción para web 55](#_Toc92802202)

[9 Funcionalidades desarrolladas 55](#_Toc92802203)

[9.1 Marco de desarrollo empleado 55](#_Toc92802204)

[9.1.1 Tiempos de desarrollo 55](#_Toc92802205)

[9.2 Prioridades (explicadas en puntos anteriores) 55](#_Toc92802206)

[9.3 Dificultades 55](#_Toc92802207)

[9.4 Detalle de las funcionalidades 56](#_Toc92802208)

[9.5 Futuros desarrollos 56](#_Toc92802209)

[9.6 Pruebas 56](#_Toc92802210)

[9.7 Prestaciones 57](#_Toc92802211)

[10 Análisis y presentación de resultados, con los principales hallazgos. 57](#_Toc92802212)

[11 Conclusiones más relevantes y aportaciones teóricas y metodológicas. 57](#_Toc92802213)

[12 Anexo Entrevistas 57](#_Toc92802214)

[13 Bibliografía 57](#_Toc92802215)

# Introducción

El presente Trabajo Fin de Máster, con el título “Plataforma online para creación y anotación de vídeo”, ha sido desarrollado por Jaime Alba Cepero en el marco del Máster de Ingeniería Informática de la Universidad de Salamanca, curso 2021/2022, y ha sido dirigido por el Dr. Roberto Therón Sánchez, catedrático de la Universidad de Salamanca.

## Agradecimientos

# Motivación

A lo largo cualquier carrera profesional, y muy especialmente dentro del campo de la tecnología, se presentan multitud de elecciones acerca de cómo se relacionará un usuario con nuestro trabajo: qué objetivo quiere conseguir, cómo quiere llegar a dicho objetivo o las alternativas que existen para conseguirlo. Priorizar la consecución de resultados tangibles hace que en muchas ocasiones se carezca de un período previo de reflexión y análisis acerca de qué es lo que espera realmente el usuario o cliente y cuál es el mejor método para conseguirlo: centrarse en el tiempo y el coste suelen comprometer la correcta definición del alcance. Aunque los modelos de trabajo ágiles han intentado paliar este problema actualizando la metodología de gestión de los proyectos, sigue existiendo un problema de raíz: el usuario final muy rara vez coincidirá (en conocimientos, aspiraciones, etc.) con los constructores de la aplicación.

Partiendo de esta reflexión, se ha elaborado en este TFM un estudio de las necesidades de los usuarios para el campo de la edición de vídeo que, aunque muy explotado, sigue ofreciendo grandes capacidades de mejora para usuarios no familiarizados con este tipo de herramientas, que tienden hacia una excesiva profesionalización. El objetivo final del trabajo es ofrecer un entorno para la creación y anotación de vídeos, que permita generar resultados con la mayor rapidez posible de manera intuitiva.

A lo largo de la investigación previa para la realización del trabajo se han analizado multitud de plataformas de edición de vídeo, todas ellas ofreciendo gran cantidad de recursos y posibilidades, muchas de ellas de manera gratuita, pero con varios problema compartido: presuponen conocimientos previos en las tareas de edición, en algunos casos tienen una curva de aprendizaje que supone una barrera en su uso, y ofrecen un abanico de funcionalidades que el usuario medio ni se plantea explorar. Existe en este tipo de productos más preocupación por atender las necesidades de los especialistas en la materia que en las de los no iniciados.

Tal y como se expone en (Watson, 2018), las aplicaciones de entretenimiento son las que son eliminadas de los dispositivos móviles con mayor velocidad (el tiempo entre el último uso y el borrado no llega a las 9 horas en iOS), de lo que se deduce que el usuario espera unas determinadas funciones y éstas resultan complicadas de alcanzar, o directamente la aplicación no ofrece lo que esperaban. Para aplicaciones web y de escritorio, cuya desinstalación no siempre se monitoriza (como puede hacerse en iOS o en Android), se considera que este fenómeno, conocido como *customer churn*, puede aplicarse en un porcentaje similar.

En este trabajo se ha trabajado sobre este fenómeno, desarrollando una aplicación de edición de vídeo que ataje los problemas detectados en la captura de necesidades y expectativas no satisfechas de los usuarios. Se hará hincapié en las distintas tecnologías disponible actualmente para el desarrollo de aplicaciones web, y en las dificultades que implican su aplicación para la consecución de un programa funcional y que elimine las barreras con el usuario.

El software reducido se podrá aplicar principalmente

CONTINUAR

El punto de partida podrán ser imágenes, textos u otros vídeos, se permitirá la inclusión de sonido y la sobreimposición de comentarios en texto, o vídeo + audio. Se pondrá especial énfasis en el manejo de timelines y en la creación WYSIWYG. El resultado será la creación de un vídeo que incluya los elementos deseados, permitiendo una exportación rápida a otras plataformas o soportes.

La plataforma está orientada para su uso en entornos como el educativo o el entretenimiento.

## Definición del tipo de trabajo

En esta memoria se presentará un trabajo individual enfocado al análisis conceptual, técnico y de mercado de la herramienta desarrollada para la edición de vídeo, indicándose los principios teóricos aplicados y su traslación al software final.

El trabajo se entrega bajo licencia [CC BY-SA 4.0 ](http://creativecommons.org/licenses/by-sa/4.0/?ref=chooser-v1) , y se fundamentará en tres pasos principales:

* Identificación de personas,
* definición de servicio y
* síntesis de funciones.

El software desarrollado consiste en una aplicación web que demostrará los diferentes principios aplicados; el código fuente se encuentra disponible tanto a través de la entrega oficial en la plataforma de la USAL como en la plataforma GitHub[[1]](#footnote-1). El código fuente se entregará tanto sin compilar como compilado como un archivo .jar, incluyendo todos los recursos estáticos necesarios; se entregará igualmente un volcado de javadoc donde se mostrarán los contenidos Java del desarrollo.

## Resumen de la propuesta inicial y comparación con el desarrollo final

En esta sección se recoge de manera resumida la propuesta inicial presentada originalmente, y se mostrarán qué elementos de dicha propuesta han sido eliminados del producto final y el motivo de dicha eliminación.

La propuesta indicaba que la entrada un vídeo, de galería del usuario u obtenido mediante captura directa. El objetivo es que se acepten varios formatos, duraciones y códecs, descartando la posibilidad de editar un vídeo existente en otras plataformas. Esta definición inicial se amplió para, no sólo permitir distintos códecs y tamaños, sino para también permitir la transformación de imágenes a vídeo (permitiendo la creación de presentaciones), pudiendo combinar ambos tipos de entradas.

### Anotaciones

Tal y como se recogía en la propuesta inicial, se ha elaborado un *layout* para la anotación de vídeos, permitiendo tanto vídeos como imágenes en formato horizontal o vertical. No obstante, y como consecuencia del proceso de investigación realizado, se ha descartado la posibilidad de anotación mediante un API que permitiera una entrada JSON o XML, centrando el software en un sistema WYSIWYG.

Las anotaciones se podrán escribir en formato libre (no viéndose limitadas a plantillas, como recogía la propuesta inicial), ofreciéndose una serie de *stickers* (imágenes prediseñadas, con iconografía naíf) que se pueden añadir además de texto en diversos tamaños y fuentes, y limitar en el tiempo. Se ha descartado el uso de subtítulos automáticos o *closed captions*, ya que, pese a encontrarse muy estandarizados, ofrecen una escasa flexibilidad y capacidad de configuración.

Otras funcionalidades como captura de webcam o transiciones han sido descartadas del alcance final por consideraciones de tiempo. Se trata de unos elementos reclamados durante las entrevistas al público objetivo, y que podrían ser añadidas en versiones posteriores de la herramienta.

### Formato de salida

Actualmente, la edición de vídeo presenta una limitación fundamental: cuando se añade contenido a un vídeo es necesaria realizar una nueva codificación de éste, que es un proceso computacionalmente costoso e impacta por tanto en la experiencia final del usuario. Por tanto, y también con el objetivo de simplificar el manejo del software para usuarios no experimentados (el público objetivo) se han limitado las posibilidades de exportación. Igualmente, se ha eliminado la opción de generación de gifs por considerarlo fuera del alcance deseado de la herramienta.

Por último, hay que indicar que se ha descartado el desarrollo de un formato propietario de anotación de vídeo (crear la anotación y que se pudiera reproducir sólo con una herramienta ad-hoc), con el propósito de maximizar la compatibilidad del resultado final.

### Versión de escritorio y versión móvil

Aunque la propuesta inicial incluía una versión tanto para navegadores web de escritorio como para navegadores en terminales móviles, el producto final se ha desarrollado con el foco exclusivo en escritorio, debido principalmente a que un desarrollo multiplataforma hubiera condicionado la capacidad de desarrollo de funcionalidades, y al haberse comprobado la escasa manejabilidad de la edición en un navegador web de terminal móvil (el desarrollo debería haberse centrado en una *app* desarrollada con Android Studio, por ejemplo).

La disposición de elementos no ha seguido tampoco la propuesta inicial, más cercana a los editores de vídeo e imagen estándar. El *layout* se centra en mostrar sólo los elementos a medida que van siendo necesarios en cada momento, con el propósito de mejorar la experiencia de usuario y evitar una sensación de sobrecarga.

### Características del entorno

De acuerdo con la propuesta inicial, se ha realizado un enfoque que prime factores como usabilidad y velocidad, en ocasiones sacrificando la calidad máxima en los vídeos generados en favor de la obtención de un resultado lo suficientemente válido en un tiempo razonable.

Queda pendiente la exportación del resultado a distintas redes sociales, tipo Twitter, Facebook, Vimeo, etc.

### Enfoque de implementación

La implementación final del proyecto se ha realizado empleando el framework *Spring Boot*[[2]](#footnote-2), que ofrece grandes capacidades fácilmente reutilizables basadas en lenguaje Java con un servidor de aplicaciones Tomcat, combinado con el motor de plantillas *Thymeleaf[[3]](#footnote-3)* (frente a las propuestas iniciales React o Angular). Al no aplicar funciones de IA, se ha descartado el uso de Python o Flask.

Por último, cabe destacar el uso del software FFMPEG[[4]](#footnote-4), que se ha comprobado que es el estándar de facto para el manejo, conversión y codificación de vídeo. Este software funcionará en la parte servidora, y requerirá una alta capacidad de cómputo.

### Principales limitaciones

A lo largo del desarrollo del software se han atajado los múltiples problemas asociados al enfoque WYSIWYG para ofrecer una experiencia consistente. Han quedado fuera del alcance final la implementación de determinados elementos de seguridad para evitar ataques tipo DoS, por ejemplo.

También ha sido necesario sobrevenir múltiples problemas con la herramienta FFMPEG que, al ofrecer un amplio abanico de opciones de recodificación y adaptación de contenidos, posee una pronunciada curva de aprendizaje.

Igualmente, el enfoque multiplataforma de la aplicación tanto en su parte cliente (navegadores web) como servidora (se ha verificado en Windows y Linux), ha supuesto un desafío para asegurar la compatibilidad de todos los elementos empleados (desde el programa FFMPEG hasta la aplicación web, la parte cliente o las configuraciones necesarias dependiendo del entorno).

## La falsa seguridad del especialista

En el capítulo 5.3.7 de este mismo documento (La barrera tecnológica y de nomenclatura) se exponen diversos conceptos técnicos que han sido manejados a lo largo del desarrollo del TFM. Se trata de terminología que puede llegar a ser confusa y compleja, pero que se maneja frecuentemente en el ámbito profesional de la imagen y el sonido.

Al igual que un pintor puede presuponer que todo el mundo conoce la composición de los pigmentos básicos, o que un abogado tiende a pensar que todos estamos familiarizados con la legislación vigente, en el mundo de la tecnología se puede llegar a presentar una sensación de que todo el mundo conoce determinados conceptos o, en el peor caso, que *les importa* siquiera: una fascinante conversación sobre las ventajas de H.265 frente a VC1 puede suponer una tortura sin paliativos para alguien que únicamente quiera juntar los vídeos de sus actuaciones del colegio: puede que acabe usando esos códecs, pero en la mayor parte de los casos no quiere “complicarse la vida”.

Pese a que hay productos excepcionalmente avanzados que son capaces de configurar, adaptar o alterar los más intrincados aspectos de una imagen o un audio, la audiencia seleccionada para este trabajo se encuentra en el otro espectro de usuarios (el usuario “ocasional”), por lo que ha sido necesario un cambio de mentalidad para favorecer la abstracción de determinados conceptos.

## Situación actual

### Generación de contenido de vídeo

Según YouTube[[5]](#footnote-5), más de 500 horas de vídeo son subidas cada minuto a su plataforma. Esto representa 720.000 horas de vídeo diariamente. El volumen de vídeo generado diariamente es astronómico, considerando que esos son los datos de una única plataforma (actualmente superada por TikTok en volumen de visualizaciones), y que son datos de vídeos compartidos en redes. El volumen puede ser aún más abrumador si consideramos vídeos compartidos de manera individual o directamente vídeos no compartidos con terceros. Es por tanto seguro decir que millones de horas de vídeo son procesadas a diario, generalmente con fines de entretenimiento.

Es de esperar que esta tendencia aumente (según Statista[[6]](#footnote-6), el 99% de los estadounidenses entre 18 y 29 años usa internet) y que la destreza en el manejo de herramientas destinadas a redes sociales sea cada vez mayor.

Sin embargo, siempre existirá un rango de entrada ¿?????????????????????????

### Productos de pago y licencias

Código libre -> poco amigable.

### Uso particular / educativo.

# Contexto

## Relación con objetivos y competencias del máster

A lo largo de la elaboración del trabajo se han detectado diversas interacciones con las materias vistas en el Máster de Informática, así como con los contenidos competenciales dentro de las mismas, entre los que destacan:

* Creación de Empresas de Base Tecnológica.
  + Análisis de la función de las tecnologías y de la innovación en la definición de estrategias y la consecución de ventajas competitivas.
  + Planificación y gestión de los procesos de I+D+i.
* Diseño, Administración e Integración de Infraestructuras TI.
  + Adquirir los conceptos básicos sobre tecnologías y modelos de integración.
  + Ser capaz de integrar tecnologías, aplicaciones, servicios y sistemas informáticos en contextos amplios y multidisciplinares.
  + Conocer estándares y tecnologías de integración de sistemas informáticos.
* Computación Gráfica.
  + Conocimiento de los fundamentos de la imagen digital.
  + Capacidad para aplicar técnicas básicas de tratamiento de la imagen y el vídeo.
  + Habilidad para el manejo de software para el tratamiento de imágenes y vídeos.
* Computación de Altas Prestaciones.
  + Conocimiento de las estrategias de diseño existentes para proyectos distribuidos, orientadas principalmente a computación de altas prestaciones.
  + Capacidad para estimar las necesidades de un sistema de cómputo de altas prestaciones y su distribución equilibrada entre los recursos disponibles.
  + Capacidad para proyectar, calcular, diseñar e implantar sistemas distribuidos de cálculo de altas prestaciones.
* Gobierno de Tecnologías de la Información.
  + Capacidad de diseñar un proyecto de consultoría o auditoría relacionado con el Gobierno de las Tecnologías de la Información.
* Paradigmas avanzados de la interacción Persona/Ordenador.
  + Capacidad para identificar las posibilidades de operación de un sistema tecnológico, posibilidades de acción de las personas que lo usan y reacciones del sistema.
  + Conocimiento de los principios y técnicas para la visualización de datos.
  + Habilidad para el trabajo con lenguajes de programación y software para la visualización de información.
  + Conocimiento de la importancia de los factores humanos y los fundamentos de la comunicación, narrativa e interacción.

## Relevancia social

Durante la elaboración del presente trabajo se ha constatado la multitud de servicios de software

Interaction. As human-technology engagement continues to increase, the end game is simplicity—transparent and natural interactions.

Enablers. Digital experience, data and analytics, and cloud have spawned multiple innovative business models and strategies.

Análisis de necesidades.

# Antecedentes tecnológicos

Tal y como se puede ver en (Donnelly, 2021), gran parte del actual proceso de edición de vídeo es heredero directo del montaje cinematográfico, originalmente cercano al “collage” en su aplicación práctica, tanto en iconografía (% para indicar un recorte, o ´ como símbolo inequívoco de “vídeo”) como en terminología (“corte” para indicar un cambio de plano, o “cuadros por segundo” para la frecuencia de fotogramas/imágenes por segundo).

En algunos casos sorprende ver cómo esta nomenclatura ha sobrevivido pese a que se ha perdido la referencia física al elemento original: por ejemplo, tanto el icono de “grabar fichero”, heredero del disco óptico de 3.5’, como el de la película fotográfica (que permite identificar el contenido en vídeo) se han conservado en los usuarios que no han visto nunca ni un diskette ni una tira de celuloide o negativo fotográfico), convirtiéndose en estándares similares al del sobre para el correo electrónico, la cámara analógica o el obturador para la captura de fotografías o el auricular para identificar las llamadas telefónicas.

## Evolución del concepto de edición

El concepto de edición originalmente en las primeras películas, que consistían en tomas únicas, y se limitaban a “unir” en un mismo rollo de celuloide distintas películas. Desde el primer corte en una película[[7]](#footnote-7) hasta la aparición de la edición no-lineal.

Destacan dos conceptos tecnológicos de la edición tradicional que persisten en el software de edición actual:

* Moviola: creado en 1924, este dispositivo permite a un editor ver la película mientras realizan el montaje de ésta (hasta ese momento, se hacía mirando manualmente los cortes).



Ilustración 1. Moviola Intercine (alrededor de 1950)

* Edición no lineal (Non-Linear Editing, NLE): proceso que permite trabajar en un clip dentro del proyecto sin afectar a los elementos posteriores o anteriores. Trabaja con una copia del contenido, y por tanto no modifica el original[[8]](#footnote-8).

La edición digital ha introducido conceptos complejos como la corrección de color, creación de previsualizaciones para efectos digitales, etc.

También cabe destacar el concepto de edición online y offline, que en el proceso de montaje tradicional se refieren no a la conexión a redes, sino al uso del material original (online) frente al trabajo con versiones de baja resolución (offline) que era necesario por la introducción de la edición no lineal. Las herramientas más potentes de edición se fundamentan en la potencia de los equipos más modernos para trabajar con ficheros en formato RAW, y con resoluciones de hasta 32k[[9]](#footnote-9).

## Comparación con otros programas

En esta sección se revisan algunos de los editores existentes actualmente en el mercado, y se ofrecen puntos de comparación con el software desarrollado en este TFM. Se considerarán factores como la facilidad de uso, el número de interacciones necesarias para conseguir un resultado básico, las opciones de exportación, etc., además de, evidentemente, el coste.

#### Libres / freemium

Actualmente la mayor parte de los programas de edición se basan en el modelo freemium: ofrecen una serie de características más o menos avanzadas durante un determinado tiempo o restringidas desde el primer momento, dejando para los usuarios de pago aquellas funciones que consideran avanzadas, como por ejemplo la exportación en determinado formato o resolución, el uso de determinados filtros de vídeo o audio o limitar la duración de los clips.

##### De escritorio

Existe una cantidad ingente de programas de edición de vídeo; se revisan a continuación algunos de ellos, y se comparan con las necesidades detectadas.

* Shotcut[[10]](#footnote-10): una propuesta de código abierto y gratuita, ofrece un amplio abanico de opciones, filtros y efectos con aspiración de alto perfil basándose en FFMPEG. Sin embargo posee un interfaz bastante confuso (emplea una terminología no demasiado clara para las distintas áreas de trabajo) y carece de la posibilidad de insertar notas de texto.
* OpenShot[[11]](#footnote-11): presenta un perfil de usuario similar al buscado en este TFM, empleando un interfaz a priori amigable y unas opciones definitivamente orientadas a la creación de vídeos lúdicos. Posee pocos requisitos de sistema y unas opciones de exportación aceptables. Sin embargo, sorprende no encontrar la posibilidad de incluir notas de texto, una función que a priori se antoja básica en cualquier editor actual.
* Lightworks[[12]](#footnote-12): se trata de un editor de grandes capacidades, destinado principalmente a un usuario experimentado. Pese a que la descarga y el uso es gratuito durante 90 días, muchas de las opciones se encuentran bloqueadas. Existen además algunas sorprendentes limitaciones en su uso:
  + En un proyecto sólo pueden importarse vídeos con un mismo framerate, lo que limita la posibilidad de emplear vídeos de distintas fuentes.
  + No especialmente intuitivo, las opciones de exportación aparecen en un submenú oculto.
* Editor de vídeo nativo de Windows: ubicado dentro del editor de fotos de Windows, ofrece unas capacidades básicas de edición de vídeo, pero limitado a la compatibilidad nativa de códecs de Windows (no permite edición con el nuevo estándar H.265). Por otro lado, la edición de texto se limita al uso de estilos y plantillas predeterminados (no es posible insertar un emoticono), y no es posible el mismo texto para varios clips (sólo puede hacerse uno a uno), o directamente no pueden añadirse varios textos en un clip. Igualmente, no se puede emplear un clip de música para varios cortes (la edición se limita a un único clip).

##### Online

El software de edición de vídeo requiere un complicado balance en el modelo gratuito o *freemium*, ya que se trata de programas que entran en el campo de la alta especialización y pueden llegar a requerir muchos recursos hardware para su ejecución (como puede ocurrir con el renderizado de imágenes de síntesis, el cálculo científico, etc.).

Poner por tanto a disposición del público general una serie de recursos (tarjetas gráficas, servidores, discos SSD) puede suponer un desembolso importante que puede no llegar a compensar económicamente, por lo que se imponen restricciones evidentes en la longitud de los vídeos a generar

* Clipchamp[[13]](#footnote-13): ofrece un editor con multitud de alternativas para la edición online. Aunque posee bastante potencia y opciones de edición, se presentan varios problemas que están solventados en la aplicación presentada:

1. No es compatible con todos los navegadores. Esta aplicación no funciona en Firefox, sólo lo hace en navegadores basados en Chromium, lo que puede despertar una sensación de rechazo hacia la herramienta (“¿por qué tengo que cambiar mi navegador?”).
2. La exportación gratuita sólo exporta hasta 480p (SD). Esta circunstancia se ha visto frecuentemente en diversos programas tipo freemium: las opciones de edición y creación son completas, pero en el momento de realizar la exportación existe alguna limitación fundamental, bloqueada para su uso de pago. Esta situación genera un escenario de frustración para un usuario medio que, tras haber dedicado varios minutos a familiarizarse con la interfaz, descubre que su trabajo ha sido en vano al no desear pagar por una herramienta que usará en contadas ocasiones.
3. Aunque posee la posibilidad de crear un vídeo desde 0, esta opción no se encuentra entre las destacadas, y se favorece el uso de plantillas (que resultan confusas de utilizar).

* Kapwing[[14]](#footnote-14): permite la edición sin uso de plantillas ni severas restricciones asociadas al plan gratuito (salvo una marca de agua), y la opción de pago no es necesaria a no ser que se deseen vídeos de gran tamaño o resolución. Ofrece varias opciones de anotación y transformación que resultan agradables y con una curva de aprendizaje no muy pronunciada. Es por lo general un gran programa en el que, sin embargo, se han detectado varios problemas durante las pruebas realizadas:

1. Emplea un sistema de capas, muy habitual en los editores de vídeo, que funciona de manera algo errática (por ejemplo, subir varios ficheros de vídeo a la vez provoca que aparezcan en el timeline superpuestos, y no consecutivos).
2. Las anotaciones se basan en capas (que no resultan intuitivas de añadir), y es necesario añadir una nueva capa por cada nota simultánea que se quiera añadir.
3. Se ha ofrecido probar esta aplicación a un miembro del público objetivo de la aplicación del TFM (niño de 12 años), y ha necesitado de varios intentos para lograr un resultado básico.
4. Tras varios intentos por editar un vídeo sencillo empleando Firefox, el software sigue mostrando un mensaje de “el contenido se está subiendo aún”. Este problema no se observa en navegadores basados en Chromium.
5. Ofrece algunas alternativas de edición de dudosa aplicación, como por ejemplo vídeos con formatos aberrantes en alto o ancho.

* Powtoon[[15]](#footnote-15): según el buscador Google, se trata de una herramienta online destinada a la creación de vídeos. Las principales críticas radican en que no aclaran realmente el propósito principal de la aplicación:

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Ilustración 2. Captura del menú principal de PowToon

* + El aspecto de la pantalla principal, una vez realizado el login vía OAuth de Google, resulta abarrotado y no resulta sencillo ubicar funciones básicas.
  + Las opciones disponibles son en su mayor parte de pago, y la opción de importación de vídeos aparece aún no disponible. Se basa fuertemente en el uso de plantillas que presuponen disposición del usuario a usarlas.

AVIDEMUX

Handbrake

VLightworks. The best free video editing software overall. ...

KineMaster. The best free video editing software for mobile. ...

OpenShot. The best free video editing software that's open source. ...

DaVinci Resolve. ...

VideoPad. ...

Windows Video Editor. ...

VSDC. ...

Hitfilm Express.

#### Pago

Adobe Premiere

Apple iMovie

CyberLink PowerDirector

Pinnacle

DaVinci resolve

### Elementos comunes de los editores

## Tendencias

(Ver artículo evolución editores de vídeo a lo largo del tiempo)

## Nicho

(Ver artículo uso de editores de vídeo)

## Objetivos y metodología

No sólo que funcione, sino que sea intuitivo.

Cookies para guardar preferencias?

# Delimitación del objeto del trabajo

## Evolución del análisis de la usabilidad

Tradicionalmente, los interfaces programas informáticos han tendido a ser diseñados por las personas que los desarrollaban. Esta circunstancia puede llegar a provocar una brecha entre el usuario y el producto que en ocasiones no se puede solventar.

Aunque la digitalización de nuestro día a día se ha consolidado hasta el punto de que nuestra interacción como personas se ha visto alterada en el curso de las últimas décadas (Harris, 2014), quienes estamos inmersos en el desarrollo tecnológico tendemos a presuponerles a todos los usuarios un nivel de conocimientos o interés mínimo en el funcionamiento interno de los dispositivos tecnológicos, cuando en realidad la convivencia con la tecnología no hace automáticamente que ésta sea entendida. Compartir cierta parte mínima del lenguaje técnico puede crear la falsa sensación de que cualquier usuario de un software conocerá la misma jerga que el desarrollador: “si sabe lo que es editor de textos, puede que entienda \hbox{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125ex X}”[[16]](#footnote-16). Ese límite es aún más confuso cuando se incrementa de forma constante el nivel de penetración del lenguaje tecnológico en nuestro día a día.

Por tanto, pueden definirse una serie de aspectos a evitar en los programas informáticos similares a los que podemos buscar en las interacciones humanas: recibir información pertinente en el momento adecuado, hablar un idioma común, o que nos faciliten determinada tarea. Es necesario saber para qué usuario se está diseñando la aplicación, qué lenguaje utiliza y qué pretende obtener de ella.

## La usabilidad en el editor de vídeo

• Un sistema es usable si los usuarios pueden hacer rápida y fácilmente sus tareas

• La usabilidad descansa en cuatro puntos:

– Una aproximación al usuario: usabilidad significa centrarse en el usuario

– Un amplio conocimiento del contexto de uso: las personas utilizan los productos para incrementar su productividad

– El producto ha de satisfacer las necesidades del usuario: los usuarios son personas ocupadas que tratan de realizar tareas

– Son los usuarios, y no los diseñadores o desarrolladores, quienes determinan cuándo un producto es fácil de usar

## Consideraciones para el éxito del producto

Según se expone en (Cooper, 2014), el éxito de un producto se sustenta en tres pilares fundamentales, cuyo enfoque se expone en este apartado. Aunque a priori se hace hincapié en la referencia a “productos digitales”, no resulta descabellado aplicar estos mismos principios a cualquier bien o servicio que se desee crear.

### Adaptación al usuario / cliente

De acuerdo con los principios del Design Thinking (Interaction Design Foundation, 2021), el primer paso para conseguir un producto exitoso consiste en saber para quién se está creando, cuáles son sus motivaciones, aptitudes y actitudes y, especialmente, cuáles son sus objetivos (Cooper, 2014).

#### Investigación de las necesidades y problemas a resolver

Partiendo de un proceso de comprensión y empatía con las necesidades del usuario objetivo del producto se podrán descartas las ideas preconcebidas del equipo que creará el producto. Durante el proceso de entrevistas y el diseño de personas se generan historias de usuario que permitirán incrementar el grado de empatía con los usuarios finales.

En el caso de este trabajo, a lo largo de su elaboración se ha podido comprobar que la edición de vídeo a nivel doméstico y académico es una necesidad recurrente, y que no siempre posee la mejor forma de ser cubierta: existe una profesionalización excesiva de las herramientas más populares, y las más sencillas no siempre cubren las expectativas.

#### Declaración de los problemas y necesidades

Mediante la destilación de la información recopilada durante la fase de investigación se podrá continuar con la creación de las *personas[[17]](#footnote-17)*. De esta manera se enumerarán los problemas específicos a resolver, ofreciendo una alternativa a los productos tradicionales, revisados en la sección 4.3 (Comparación con otros programas).

Los problemas declarados (*problem statement*)

El detalle de la creación de personas se encuentra en la sección 6.3, donde se incluyen las historias de usuario que han permitido destilar las necesidades reales a desarrollar.

It’s time to accumulate the information gathered during the Empathize stage. You then analyze your observations and synthesize them to define the core problems you and your team have identified. These definitions are called problem statements. You can create personas to help keep your efforts human-centered before proceeding to ideation.

#### Fase de ideación

Challenge Assumptions and Create Ideas

Now, you’re ready to generate ideas. The solid background of knowledge from the first two phases means you can start to “think outside the box”, look for alternative ways to view the problem and identify innovative solutions to the problem statement you’ve created. Brainstorming is particularly useful here..

#### Prototipado

A partir de la información recogida en fases anteriores, se creó un modelo inicial de baja resolución para poder definir las funciones básicas y entender las necesidades de los usuarios.

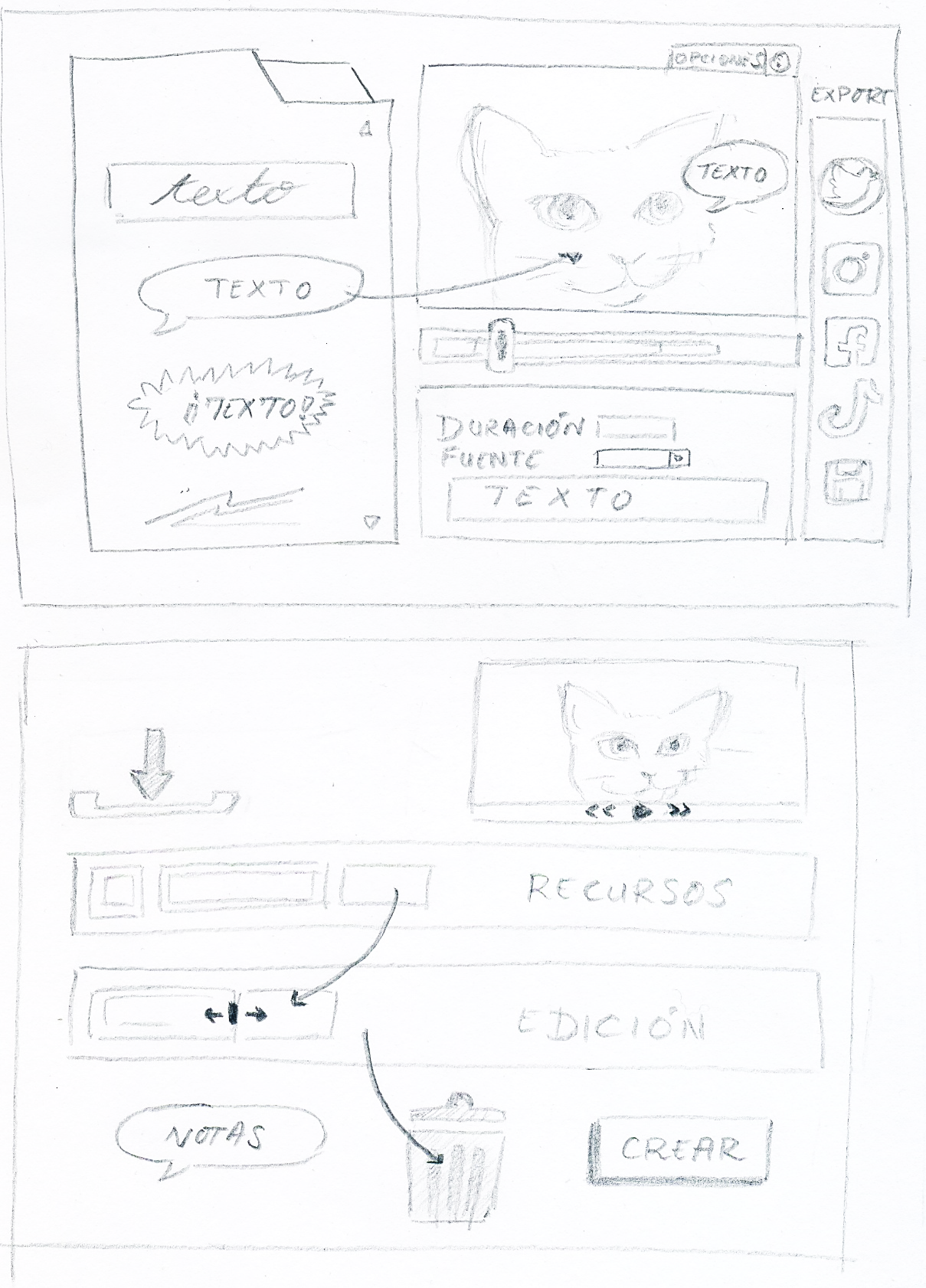


Ilustración 3. Primer prototipo, presentado en la propuesta original del TFM

Este primer prototipo, presentado en la propuesta original del TFM, pasó por varias iteraciones y revisiones a lo largo del desarrollo, considerando varios factores:

* El diseño sigue (en la medida de lo posible) un flujo natural de arriba a abajo, es decir, la primera fase del trabajo (importación de recursos) se presenta en la zona superior izquierda, la biblioteca en la zona media (donde se ve la disponibilidad y la duración original de cada clip), y la zona de trabajo en la zona inferior, incluyendo la activación de notas, la papelera (para eliminar los clips de la zona de edición) y el botón de generación.
* Las opciones se van presentando a medida que se necesitan. Se propone por tanto un paradigma diferente al de herramientas estándar, por ejemplo, la herramienta Kapwing presenta este layout por defecto:

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 4. Disposición por defecto de la herramienta Kapwing

El prototipo revisado ofrece un nuevo enfoque en el que se mostrarán las opciones de manera más orgánica y a medida que el flujo de trabajo las vaya requiriendo. Por ejemplo, la fase de importación sólo mostraría las opciones de importación, las de edición de notas sólo las opciones de edición de notas, etc. De esta manera se pretende conseguir una experiencia menos agresiva, y ofreciendo un número similar de operaciones finales:

Diagrama

Descripción generada automáticamente

Ilustración 5. Prototipo revisado

Con este segundo prototipo se pretende obtener una plantilla a partir de la cual diseñar el producto con la experiencia de usuario como foco principal.

#### Prueba de la solución adoptada

Evaluators rigorously test the prototypes. Although this is the final phase, design thinking is iterative: Teams often use the results to redefine one or more further problems. So, you can return to previous stages to make further iterations, alterations and refinements – to find or rule out alternative solutions.

Overall, you should understand that these stages are different modes which contribute to the entire design project, rather than sequential steps. Your goal throughout is to gain the deepest understanding of the users and what their ideal solution/product would be.

Three types of scenarios The Goal-Directed Design method employs three types of persona-based scenarios at different points in the design process, each with a successively more interface-specific focus. The first—the context scenario—is used to explore, at a high level, how the product can best serve the needs of the personas. The context scenarios are created before any design sketching is performed. They are written from the persona’s perspective, focusing on human activities, perceptions, and desires. It is when developing this kind of scenario that the designer has the most leverage to imagine an ideal user experience. More details about creating this type of scenario can be found in the section “Step 4: Construct con- text scenarios.” Once the design team has defined the product’s functional and data elements and devel- oped a Design Framework (as described in Chapter 5), a context scenario is revised. It becomes a key path scenario by more specifically describing user interactions with the product and by introducing the design’s vocabulary. These scenarios focus on the most significant user interactions, always paying attention to how a persona uses the product to achieve its goals. Key path scenarios are iteratively refined along with the design as more and more detail is developed. Throughout this process, the design team uses validation scenarios to test the design solution in a variety of situations. These scenarios tend to be less detailed and typically take the form of a number of what-if questions about the proposed solutions. Chapter 5 covers development and use of key path and validation scenarios.

### Modelo de negocio

El modelo de negocio de este tipo de aplicaciones está basado principalmente en experiencias freemium, o gratuitas con un período de prueba. La abrumadora mayoría de las soluciones gratuitas analizadas para este trabajo (vistos en la sección 4.3)

xxxxxxxxxxxxxxxxxxxxxxxxx

### Competencia tecnológica

### Comentarios

No vertical / aislado -> modelo ágil

Publicidad

Sostenibilidad como startup

Capacidad de procesamiento / almacenamiento

Ver requisitos hardware (7.4)

# Diseño metodológico

A lo largo de esta sección se expondrán los diferentes mecanismos utilizados para perfilar a los usuarios objetivo, con el propósito de ajustar el desarrollo del producto a sus necesidades y objetivos.

## Búsqueda de necesidades

A partir de la propuesta inicial, se ha realizado una serie de entrevistas con grupos de distintas edades (primer grupo entre 10 y 14 años, y segundo grupo de entre 35 y 45 años), con un sólido perfil tecnológico para sus respectivas franjas de edad.

Tests de usabilidad

### Entrevistas

Encuestas.

Entrevistas

#### Resultados

##### Experiencia previa de los entrevistados

La mayoría de los usuarios ya habían utilizado, en mayor o menor medida, editores de vídeo:

* iMovie para iPad

##### Requisitos básicos capturados

Lista de funcionalidades de un editor de vídeo básico e importancia por grupo de usuarios

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidad** | **Puntuación grupo 1** | **Puntuación grupo 2** | **Puntuación grupo 3** |
| Fácil importación / exportación |  |  |  |
| Ajustar duración de los vídeos |  |  |  |
| Añadir fotos |  |  |  |
| Añadir música |  |  |  |
| Añadir stickers |  |  |  |
| Añadir texto |  |  |  |
| Crear presentación |  |  |  |
| Transiciones |  |  |  |
| Rotar |  |  |  |
| Añadir filtros |  |  |  |
| Ajustar exposición o contraste |  |  |  |
| Edición por capas |  |  |  |
| Distintos formatos de salida |  |  |  |
| Edición no lineal |  |  |  |
| Preview |  |  |  |

for both absolute newbies and the many possible cases

an expert might encounter. But don’t let this business requirement influence your design

thinking. Yes, you must provide those features for expert users. Yes, you must support

those transient beginners. But in most cases, you need to apply the bulk of your talents,

time, and resources to designing the best interaction possible for your most represen-

tative users: the perpetual intermediates. When digital products follow the principle

of commensurate effort, the learning curve doesn’t go away, but it disappears from the

user’s mind—which is just as good.

### Tiempo medio de uso de la aplicación

Ante el enorme abanico de alternativas informáticas que existe para la realización de múltiples tareas, en muchas ocasiones el *churn* es la consecuencia lógica que se produce cuando un usuario requiere realizar una tarea poco habitual para él, que quiere que le lleve poco tiempo, y en la que no está dispuesto a asumir una curva de aprendizaje, por mínima que sea ésta.

Tal y como se ha expuesto anteriormente en este documento, se ha diseñado una aplicación con la que el usuario podrá pasar el tiempo mínimo imprescindible, reduciendo en la medida de lo posible la curva de aprendizaje necesaria para realizar las tareas (generalmente, juntas varios vídeos y añadir varias etiquetas).

Se juega por tanto con un concepto que podría denominarse “compromiso relajado” o “*loose engagement”*, una destilación del “*customer engagement*” en la que se ofrece el número justo de funciones que espera el público target de manera lo más sencilla posible con el objetivo de que usen la aplicación el menor tiempo posible, pero que la recuerden y la vuelvan a usar cuando vuelva a ser necesaria. Se presupone por tanto un usuario impaciente que espera obtener su resultado lo antes posible, y que no quiere responder a encuestas de satisfacción sobre nuevas funcionalidades, suscribirse a los boletines de novedades o formar parte de la comunidad de creadores. Aunque estos perfiles (con voluntad de comprometerse a medio o largo plazo con el producto) ciertamente existen, no siempre se presta atención a los usuarios que “no quieren comprar nada, sólo están mirando”.

### Dispositivos de visualización

(Entrevistas)

### Uso que se dará

## Diseño de personas

Con el objetivo de completar un correcto perfil de los usuarios de la aplicación, se ha definido un total de 3 personas siguiendo los paradigmas del *design thinking*.

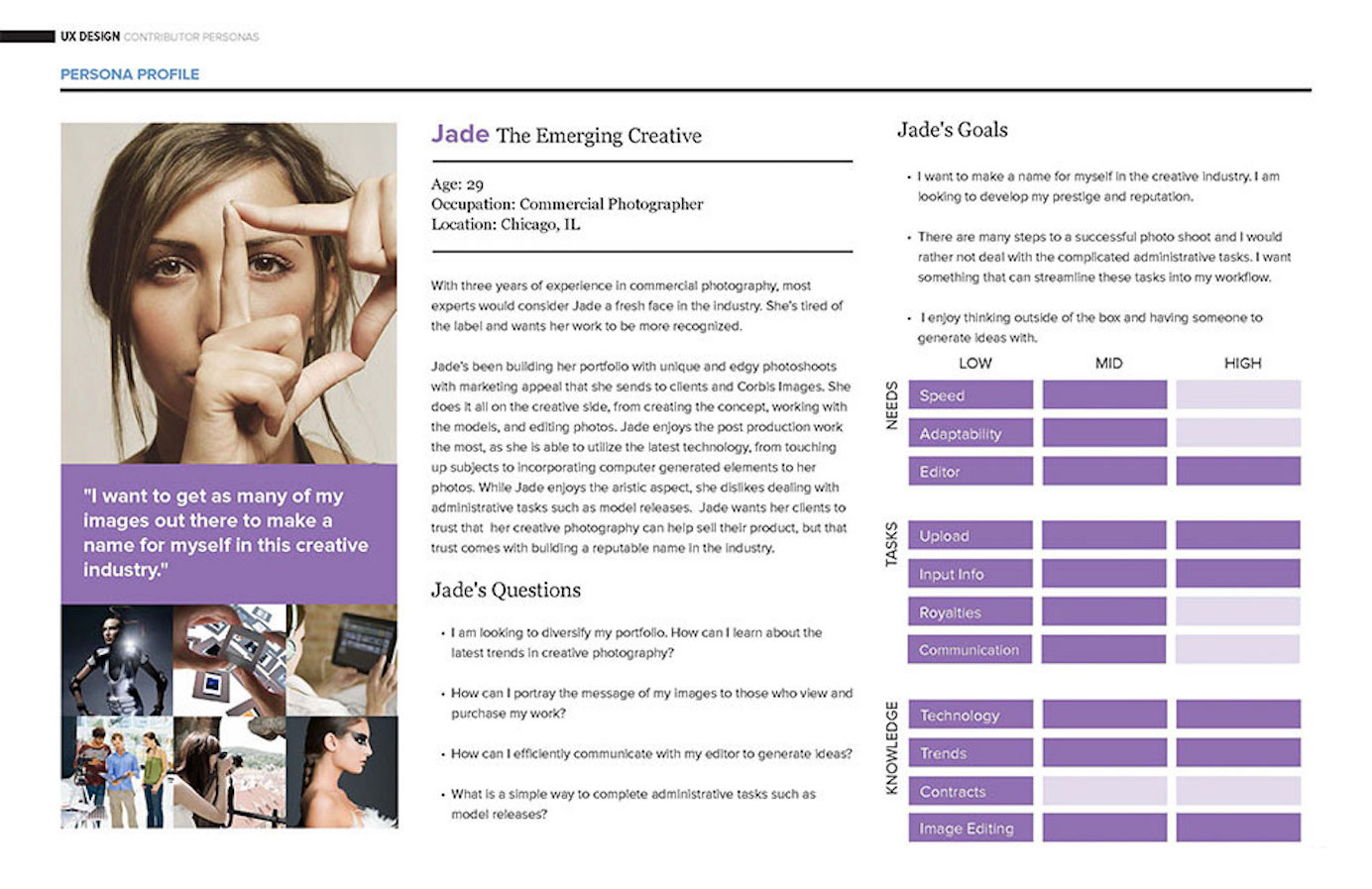
1 actor principal -> niño – objetivo -> crear un vídeo a imagen de los que se generan para youtube.

2 actores secundario -> persona con conocimiento informático, pero sin interés en terminología de vídeo / persona sin conocimiento informático.

1 actor terciario -> persona con conocimiento de herramientas de vídeo.

Incluir historias de usuario

<https://www.nngroup.com/articles/why-personas-fail/>



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [[18]](#footnote-18)      **Teresa**  Estudiante aplicada  **Edad**:13 años.  **Ocupación**: estudiante  **Educación**: ESO  Lugares de conexión:  Equipo:  Uso diario de internet: | **Uso de herramientas de vídeo**  **Interacción**  **Principales influencias**  Asdf  Sdf | | **Objetivos y motivación**  **Preguntas que se hace**  **Historia personal** | |
| [[19]](#footnote-19)    **Raúl**  Funcionario local  **Edad**:  **Ocupación**:  **Educación**:  Lugares de conexión:  Equipo:  Uso diario de internet: | | **Uso de herramientas de vídeo**  **Interacción**  **Principales influencias**  Asdf  Sdf | | **Objetivos y motivación**  **Preguntas que se hace**  **Historia personal** |

|  |  |  |
| --- | --- | --- |
| [[20]](#footnote-20)  Hombre sentado en una cama  Descripción generada automáticamente con confianza baja  **Raúl**  Funcionario local  **Edad**:  **Ocupación**:  **Educación**:  Lugares de conexión:  Equipo:  Uso diario de internet: | **Uso de herramientas de vídeo**  **Interacción**  **Principales influencias**  Asdf  Sdf | **Objetivos y motivación**  **Preguntas que se hace**  **Historia personal** |

## Consideraciones de diseño

Para la elaboración del software asociado a este TFM se han considerado los siguientes factores a la hora de realizar el diseño de la usabilidad.

### 242 Part II: Making Well-Behaved Products

Figure 10-2: Adobe Creative Suite applications all make similar

use of progressive disclosure to tame the complexity of their tool

palettes for intermediates. Experts can expand them and take

advantage of the sticky expansion state, which is remembered

across sessions.

organizing for inflection

In general, controls and displays should be organized in an interface according to three

attributes: frequency of use, degree of dislocation, and degree of risk exposure.

• Frequency of use means how often the controls, functions, objects, or displays are

used in typical daily patterns of use. Items and tools that are most frequently used

(many times a day) should be immediately in reach. Less frequently used items, used

perhaps once or twice a day, should be no more than a click or two away. Other items

can be two or three clicks away. Rarely used facilities shouldn’t be removed from the

product if they provide real benefits to your personas, but they should be removed from

the everyday work space.

• Degree of dislocation refers to the amount of sudden change in an interface or in the

document/information being processed by the application caused by the invocation of

a specific function or command. Generally speaking, it’s a good idea to put these types

of functions deeper into the interface. • Degree of risk exposure deals with functions that are irreversible or may have other

dangerous ramifications. Missiles require two humans turning keys simultaneously

on opposite sides of the room to arm them. As with dislocating functions, you want

to make these types of functions more difficult for your users to stumble across. The

greater the ramifications, the more care you should take in exposing the function.

### Impacto de primer uso y facilidad de descubrimiento

El principal factor que se ha tenido en cuenta en el diseño ha sido la sencillez de uso para nuevos usuarios. Asociado al uso intuitivo de la herramienta.

### Minimalismo

Uno de los objetivos a conseguir era la de crear una falsa sensación de simplicidad. Pese a que el proceso resulta técnicamente complejo (ver secciones 6 y 7), se han condensado todas las funciones en una única página, de manera que el usuario no perderá nunca la referencia del proyecto que está realizando. Siguiendo el principio de “menos es más”, el producto se ha diseñado para requerir el menor número de interacciones para conseguir un resultado partiendo de 0. Se ha evitado el uso de submenús o pantallas adicionales en la medida de lo posible, existiendo únicamente las siguientes secciones funcionales:

* Carga y ordenación de clips de vídeo e imágenes: se trata de la sección principal, y en ella se cargarán o recuperarán los clips e imágenes del proyecto para poder realizar las acciones de edición deseadas. Una vez realizados los posibles recortes y la gestión de anotaciones (las dos secciones explicadas a continuación), el botón de generación, visible en todo momento, permitirá iniciar el proceso automático de creación del vídeo final.
* Recorte de clips.
* Gestión de notas.

### Orquestación y disposición de elementos

La aplicación se ha diseñado con un patrón arriba-abajo e izquierda-derecha. Las zonas asociadas a las distintas acciones se han ordenado siguiendo el sentido de lectura occidental: los elementos iniciales se sitúan arriba (ordenados de izquierda a derecha), mientras que a medida que se progresa en la creación se desciende por la pantalla para finalizar abajo a la derecha.

La aplicación se ha diseñado para

Pensado para alguien que no ha utilizado una herramienta de edición de vídeo nunca.

* First-time use and discoverability—Are common items easy for new users to find? Are instructions clear? Are instructions necessary?
* Cómo se ha diseñado: secuencial -> el final del ciclo está abajo a la derecha. Diseño en cascada. • Organization—Is information grouped into meaningful categories? Are items located in the places customers might look for them?
* Uso de nombres y descripciones : • Naming—Do section/button labels make sense? Do certain words resonate better than others?
* Colores e iconos.
* Effectiveness —Can customers efficiently complete specific tasks? Are they making missteps? Where? How often?

what to test

Because the findings of usability testing are often quantitative, usability research is

especially useful in comparing specific design variants to choose the most effective solu -

tion. Customer feedback gathered from usability testing is most useful when you need

to validate or refine particular interaction mechanisms or the form and expression of

specific design elements.

Usability testing is especially effective at validating the following:

### La barrera tecnológica y de nomenclatura

A la hora de comenzar a trabajar con cualquier disciplina que requiera una cierta especialización, en este caso el software de edición de vídeo y de imagen, aparece una nomenclatura básica que queda alejada del vocabulario habitual. Nuestras personas objetivo desconocerán el significado real de todos estos conceptos, que entra para ellos en el campo de la “jerga tecnológica”, y por tanto se ha tomado una serie de decisiones que, aunque comprometen la flexibilidad que pueden ofrecer otras herramientas, facilitan el proceso de creación y manejo (por ejemplo, se ha observado que un usuario no especializado, ante la posibilidad de elegir la resolución de salida, elegirá siempre la mayor, sin necesariamente ser la más práctica).

Se presenta en esta sección una serie de conceptos que han tenido que ser manejados a lo largo de la elaboración del TFM, y que se consideran particulares de este tipo de tecnología, así como las soluciones que se han adquirido con el objetivo de hacer “caja negra” conceptual.

#### Tamaño de vídeo

Dimensión, medida en píxeles, de una imagen o vídeo. La herramienta FFMPEG[[21]](#footnote-21), que ha sido la empleada en el desarrollo del código, reconoce las siguientes proporciones (*aspect ratio*) estándar:

Tabla 1. Formatos de vídeo estandarizados para FFMPEG

| **Nombre** | **AnchoxAlto (píxeles)** | **Nombre** | **AnchoxAlto (píxeles)** | **Nombre** | **AnchoxAlto (píxeles)** |
| --- | --- | --- | --- | --- | --- |
| ntsc | 720x480 | pal | 720x576 | qntsc | 352x240 |
| qpal | 352x288 | sntsc | 640x480 | spal | 768x576 |
| film | 352x240 | ntsc-film | 352x240 | sqcif | 128x96 |
| qcif | 176x144 | cif | 352x288 | 4cif | 704x576 |
| 16cif | 1408x1152 | qqvga | 160x120 | qvga | 320x240 |
| vga | 640x480 | svga | 800x600 | xga | 1024x768 |
| uxga | 1600x1200 | qxga | 2048x1536 | sxga | 1280x1024 |
| qsxga | 2560x2048 | hsxga | 5120x4096 | wvga | 852x480 |
| wxga | 1366x768 | wsxga | 1600x1024 | wuxga | 1920x1200 |
| woxga | 2560x1600 | wqsxga | 3200x2048 | wquxga | 3840x2400 |
| whsxga | 6400x4096 | whuxga | 7680x4800 | cga | 320x200 |
| ega | 640x350 | hd480 | 852x480 | hd720 | 1280x720 |
| hd1080 | 1920x1080 | 2k | 2048x1080 | 2kflat | 1998x1080 |
| 2kscope | 2048x858 | 4k | 4096x2160 | 4kflat | 3996x2160 |
| 4kscope | 4096x1716 | nhd | 640x360 | hqvga | 240x160 |
| wqvga | 400x240 | fwqvga | 432x240 | hvga | 480x320 |
| qhd | 960x540 | 2kdci | 2048x1080 | 4kdci | 4096x2160 |
| uhd2160 | 3840x2160 | uhd4320 | 7680x4320 |  |  |

La propia herramienta reconocerá vídeos en otros formatos que no respondan a estos estándares. Por ejemplo:

* 4:3 o 8:6 para 1080p 4:3 (1440 x 1080 px).
* 21:9 para formato Ultrawide HD (2560 x 1080 px).
* 2:1[[22]](#footnote-22) o 18:9 para Full HD+ (2160 × 1080 px).
* 8:5 o 16:10 en resolución 1728 × 1080 px.
* 3:2 o 15:10 para resolución 1620 x 1080 px.

El principal problema tecnológico que se puede presentar es la posibilidad de juntar en un único vídeo entradas en distintos formatos, ya no sólo en resoluciones no estándar sino directamente formatos vertical y horizontal. De la misma manera, la velocidad de procesamiento se ve seriamente comprometida a medida que se aumenta el tamaño en píxeles del vídeo de salida.

Igualmente, el usuario tipo del producto buscará un formato de conveniencia: se han recogido frases como “da igual”, “el que mejor se vea” o “el más grande” (tal y como se expondrá en las secciones Captura de necesidades y Diseño de personas) durante las entrevistas realizadas. De hecho, salvo que se desee un producto audiovisual con unas determinadas características técnicas, un gran porcentaje de usuarios (incluyendo los que conozcan la terminología, pero quieran un resultado rápido) querrá que su vídeo posea una calidad suficiente y que lo admitan los reproductores de vídeo por defecto de sus equipos (ya sea SmartTV, teléfono, navegador, reproductor de Windows, etc.).

Considerando estos factores, y con el objetivo de ofrecer una experiencia de usuario satisfactoria, se ha adquirido como compromiso una salida en horizontal hd720 (16:9), con el propósito de balancear el tamaño en píxeles con la velocidad de procesamiento del vídeo generado.

#### Pillarboxing y vídeo vertical

Este concepto es una evolución del punto anterior, pero debido a su relevancia se expone en esta sección dedicada.

Tradicionalmente, y siguiendo la fisionomía de la visión humana, las imágenes cinematográficas o de vídeo han sido creadas y reproducidas en formato horizontal. Sin embargo, la proliferación de terminales móviles (cuya orientación, siguiendo la fisionomía de la mano, es preferiblemente vertical) ha disparado el uso de vídeo vertical hasta el punto de que multitud de plataformas de vídeo (como TikTok) indican como formato preferido el vertical[[23]](#footnote-23).

El proceso de creación de usuarios para nuestra aplicación arroja que el uso principal que se le dará será el educativo o el personal, por lo que no se ha considerado en la versión entregada la posibilidad de salida vertical, y toda salida será generada con formato horizontal 16:9 (aunque sí se podrán trabajar con una combinación de ambos). Por tanto, serán vídeos que se podrán reproducir en una televisión, un portátil (en la aplicación YouTube, por ejemplo) o un proyector sin necesidad de adaptaciones adicionales.

Cuando en la aplicación, el usuario cargue un vídeo vertical, se realizará un proceso de adaptación a horizontal, transformando el vídeo de manera que quede centrado y con sendas barras negras a los laterales. Se podrán utilizar estas barras como zonas de anotación:

Forma

Descripción generada automáticamente

Ilustración 8. Conversión de vídeo vertical a horizontal mediante pillarboxing

#### Códecs de vídeo y audio

Los códecs son algoritmos para la compresión y el tratamiento, en este caso de imágenes y de audio, que se utilizan en los distintos dispositivos digitales para la captura e interpretación de la información. Por lo general, la calidad del códec se mide según la cantidad de información que pierde respecto al original y por el tamaño del archivo generado. Cuanto más complejo sea un códec, mayor será la necesidad de recursos hardware para su correcto tratamiento.

Actualmente la especificación de FFMPEG soporta hasta 13 códecs de video[[24]](#footnote-24) de entrada, incluyendo los más utilizados actualmente: H.264 (a través de la librería libopenh264), H.265/HEVC (mediante libx265), y permite la salida hasta en 24 formatos. La aplicación desarrollada permite la entrada de vídeo con cualquiera de esos códecs, aunque ajusta la salida al H.264 (AVC) que es el estándar de facto en la mayor parte de reproductores, terminales móviles y navegadores web.

Audio

#### Formatos

Los distintos códecs comentados en el punto anterior se refieren a los algoritmos de codificación y decodificación, pero existen a nivel físico (fichero) distintos contenedores en los que pueden almacenarse[[25]](#footnote-25). Cada contenedor poseerá capacidades adicionales a las propias del códec, como por ejemplo la posibilidad de añadir subtítulos, conjugar distintos códecs o menús para el vídeo.

Tabla 2. Relación los de códecs de vídeo más populares en web y sus formatos de fichero

| **Denominación del códec** | **Nombre completo** | **Contenedor** |
| --- | --- | --- |
| AV1 | AOMedia Video 1 | MP4, WebM |
| AVC (H.264) | Advanced Video Coding | 3GP, MP4 |
| H.263 | H.263 Video | 3GP |
| HEVC (H.265) | High Efficiency Video Coding | MP4 |
| MP4V-ES | MPEG-4 Video Elemental Stream | 3GP, MP4 |
| MPEG-1 | MPEG-1 Part 2 Visual | MPEG, QuickTime |
| MPEG-2 | MPEG-2 Part 2 Visual | MP4, MPEG, QuickTime |
| Theora | Theora | Ogg |
| VP8 | Video Processor 8 | 3GP, Ogg, WebM |
| VP9 | Video Processor 9 | MP4, Ogg, WebM |

La aplicación desarrollada empleará un formato de salida MP4 con códec H.264, buscando la mayor compatibilidad y calidad, y siempre con el objetivo de minimizar las decisiones que tiene que tomar el usuario, considerando su perfil medio/bajo en el manejo de esta tecnología.

st", "-c:a", "aac", "-b:a", "128k", "-threads", "8",

#### Profundidad de color

yuv420p

#### Muxer

El *muxer* toma los datos de salida del codificador, generalmente fotograma a fotograma codificado, y los empaqueta en el formato de contenedor especificado (es decir, toma la salida por ejemplo de los fotogramas en H.264 y los vuelca en un fichero MP4).

Muxers take encoded data in the form of AVPackets and write it into files or other output bytestreams in the specified container format.

The main API functions for muxing are avformat\_write\_header() for writing the file header, av\_write\_frame() / av\_interleaved\_write\_frame() for writing the packets and av\_write\_trailer() for finalizing the file.

At the beginning of the muxing process, the caller must first call avformat\_alloc\_context() to create a muxing context. The caller then sets up the muxer by filling the various fields in this context:

The oformat field must be set to select the muxer that will be used.

Unless the format is of the AVFMT\_NOFILE type, the pb field must be set to an opened IO context, either returned from avio\_open2() or a custom one.

Unless the format is of the AVFMT\_NOSTREAMS type, at least one stream must be created with the avformat\_new\_stream() function. The caller should fill the stream codec parameters information, such as the codec type, id and other parameters (e.g. width / height, the pixel or sample format, etc.) as known. The stream timebase should be set to the timebase that the caller desires to use for this stream (note that the timebase actually used by the muxer can be different, as will be described later).

It is advised to manually initialize only the relevant fields in AVCodecParameters, rather than using avcodec\_parameters\_copy() during remuxing: there is no guarantee that the codec context values remain valid for both input and output format contexts.

The caller may fill in additional information, such as global or per-stream metadata, chapters, programs, etc. as described in the AVFormatContext documentation. Whether such information will actually be stored in the output depends on what the container format and the muxer support.

When the muxing context is fully set up, the caller must call avformat\_write\_header() to initialize the muxer internals and write the file header. Whether anything actually is written to the IO context at this step depends on the muxer, but this function must always be called. Any muxer private options must be passed in the options parameter to this function.

The data is then sent to the muxer by repeatedly calling av\_write\_frame() or av\_interleaved\_write\_frame() (consult those functions' documentation for discussion on the difference between them; only one of them may be used with a single muxing context, they should not be mixed). Do note that the timing information on the packets sent to the muxer must be in the corresponding AVStream's timebase. That timebase is set by the muxer (in the avformat\_write\_header() step) and may be different from the timebase requested by the caller.

Once all the data has been written, the caller must call av\_write\_trailer() to flush any buffered packets and finalize the output file, then close the IO context (if any) and finally free the muxing context with avformat\_free\_context().

### Integración y diseño visual

Se han realizado las siguientes consideraciones en lo relativo al diseño visual de la aplicación, en respuesta a las demandas de la elaboración de perfiles y creación de personas.

#### Áreas de carga y edición

Se han empleado colores poco agresivos para diferenciar las diferentes áreas de edición, con la excepción de la zona de papelera, a la que se le ha aplicado un color más oscuro para realizar una clara diferenciación de que los clips que se arrastren ahí serán eliminados.

Por otro lado, se ha conservado la misma forma en las diferentes secciones con el objetivo de crear un aspecto visual uniforme y consolidado en toda la aplicación.

El tamaño se ha optimizado para que permita la expansión con scroll en caso de que se carguen multitud de clips.

#### Iconos de clips

Se ha empleado la iconografía Bootstrap, que mediante líneas claras ofrece una gran capacidad de abstracción

#### Opciones de edición de notas

#### Opciones de audio

### Complejidad de diseño vs. complejidad de uso

Una de las principales reflexiones que incluye este trabajo hace referencia a la relación inversamente proporcional entre simplicidad de uso y simplicidad de elaboración de un producto software. La obtención de un resultado lo suficientemente robusto es más compleja cuantas más posibilidades de uso posea el usuario final, a lo que hay que añadir factor de la simplicidad de uso final.

A mayor número de funcionalidades que se quieren ofrecer con una menor complejidad, aumenta la dificultad de implementación y mantenimiento de la aplicación; de la misma manera, la importancia de una correcta evaluación de los usuarios objetivo y sus necesidades, así como el conocimiento técnico del campo de aplicación, es mayor cuanto mayor es el número de funcionalidades que se quiere ofrecer con una menor complejidad:

Diagrama

Descripción generada automáticamente

Ilustración 6. Relación entre facilidad de uso, total de funcionalidades, importancia del conocimiento previo y capacidad de rediseño

En la ilustración previa, se considera que todas las aplicaciones que se deseen crear deberán estar dentro del área coloreada. Llegará un punto en la creación del producto que no será posible añadir más funcionalidades para una cierta facilidad de uso, y será necesario comprometer alguno de los dos factores.

Por tanto, para desarrollar cualquier producto con estos parámetros es necesario primero determinar cuáles serán sus funciones y cuál es la facilidad de uso que queremos otorgarle: si el producto a desarrollar posee muchas funcionalidades será muy complicado darle una gran facilidad de uso.

Diagrama

Descripción generada automáticamente

Ilustración 7. Ejemplo de factores que afectan a la creación de un producto.

En el caso de la herramienta de edición desarrollada para este TFM, ha llegado un punto en el que ha sido necesario comprometer la facilidad de uso y el minimalismo en el diseño con el objetivo de ofrecer un conjunto de características básicas que han sido recopiladas durante la creación de los perfiles de personas. Puede verse claramente este problema en la edición de notas dentro de la herramienta, donde se ha recurrido al uso de convenciones ligadas a los editores de texto con el objetivo de permitir acciones (como poner un texto en negrita o en cursiva) que han sido extraídas del diseño de personas.

## Diseño software

Validación w3c

Cliente / servidor

Peticiones REST

Javascript + CSS en cliente

Java + FFMPEG en servidor.

## Pruebas de usabilidad

3 programas : prototipo vs. Davinciresolve vs videopad vs. handbrake

## Evaluación del producto

### Inspección (inspection)

Unos evaluadores inspeccionan o

examinan aspectos relacionados con la

usabilidad de la interfaz

• Los inspectores de la usabilidad pueden

ser:

– especialistas en usabilidad

– consultores de desarrollo de software con

experiencia en guías de estilo de interfaces

– usuarios finales con conocimientos del dominio

• Métodos de inspección más importantes:

– Evaluación heurística

– Recorridos cognitivos

– Inspección de estándares

### Indagación (assestment)

La información acerca de los gustos y necesidades

del usuario y la identificación de requisitos es

indispensable en una etapa temprana del

desarrollo

• Se trabaja hablando con los usuarios,

observándolos, usando el sistema en el trabajo

real, obteniendo respuestas a preguntas

verbalmente o por escrito

Métodos de indagación:

– Observación de campo: • Los ingenieros en factores humanos van al lugar de trabajo de usuarios representativos y los observan trabajando para entender cómo están utilizando el sistema para lograr sus tareas y qué clase de modelo mental tienen sobre él

• Este método se puede utilizar en las etapas iniciales

del desarrollo y en la etapa de prueba del

producto

– Grupos de discusión dirigidos (focus

groups)

– Estudio de campo proactivo

– Entrevistas

– Cuestionarios

– Grabación del uso (logging)

– Card Sorting

### Test

Pruebas unitarias

Pruebas sistema

Pruebas integración

• tiempo para completar una tarea

tiempo consumido en menús de navegación

tiempo consumido en ayuda en línea

tiempo en buscar información en un manual

tiempo invertido en recuperarse de errores

• número de opciones de menú erróneos

número de opciones incorrectas en cajas de dialogo

número de selección de iconos incorrectos

número de teclas de función mal seleccionadas

• número de llamadas a la ayuda

número de pantallas de ayuda en línea

número de veces que se consulta el manual

• observaciones de frustración

observaciones de confusión

observaciones de satisfacción

– facilidad de uso del producto

– facilidad de aprender el producto

– facilidad de hacer una determinada tarea

– facilidad de instalar el producto

– facilidad de encontrar información en el manual

– facilidad de comprender la información

– utilidad de los ejemplos de ayuda

– Preferencia sobre una versión previa del producto

– Preferencia sobre un producto de la competencia

– Preferencia sobre la forma actual de hacer las tareas

### Evaluación de rendimiento

Pruebas de prestaciones.

Tabla

Descripción generada automáticamente

## Consideraciones de compatibilidad

Uno de los problemas más sorprendentes que se han detectado ya incluso antes de la elaboración de este trabajo en lo relativo a la compatibilidad es la incapacidad del sistema operativo Windows de reproducir por defecto archivos codificados con h.265.

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 9. Efecto de la incompatibilidad de Windows 11 con H.265.

Como puede verse en la Ilustración 4, el sistema operativo Windows 11 (y por extensión, cualquiera de los sistemas de Microsoft) no soporta por defecto los códecs del tipo H.265 (denominados HEVC), que es el estándar actual de compresión de vídeo de alta calidad. La abrumadora mayoría de los vídeos creados hoy en día son realizados con terminales móviles, que comenzarán a utilizar por defecto el HEVC en Android 12, lo que puede llegar a producir una situación en la que el usuario medio no pueda ver (ni editar) sus vídeos en el software nativo de Windows.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Ilustración 10. Cuota de SO en equipos de escritorio y portátiles (fuente StatCounter)

Aunque esta situación se puede

<https://microsofters.com/183001/como-conseguir-el-codec-hevc-gratis-en-windows-10-windows-11/>

Códecs en el ordenador -> h.265 en móviles pero no en navegadores o en el ordenador.

Códect en el navegador

Formatos de salida

Formatos de vídeo h.265

Transparencia

Usuarios representativos realizan sus tareas usando

el sistema (o un prototipo) y los evaluadores

observan los resultados

• Tipos de métodos:

– Medida de prestaciones

– Test remoto

– Pensando en voz alta

– Interacción constructiva

– Test retrospectivo

– Método del conductor

## Gestión de proyectos

## Línea de trabajo waterfall vs scrum

Escala de tiempo

Descripción generada automáticamente

# Relación de dependencias

## Objetivo: emplear las menores dependencias posibles

## El caso FFMPEG

## Multiplataforma

## Requisitos hardware

<https://streaminglearningcenter.com/wp-content/uploads/2019/05/SME-2019-FFmpeg-Hardware.pdf>

https://blog.wirelessmoves.com/2021/11/ffmpeg-comparison-from-nvidia-to-raspberry-pi-4.html

## Requisitos software

Por qué no se usa base de datos -> vídeo vs. Bases de datos relacionales.

# Tecnologías involucradas

## Entorno

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

Maven project

Spring boot version 2.5.6

Java

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Jar – java 11

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Cliente

## Servidor

### Máquinas de desarrollo y despliegue

#### Win

#### MV linux local

#### MV linux remota

## Interacción para web

* Multinavegador
* Codecs independientes del cliente.

# Funcionalidades desarrolladas

## Marco de desarrollo empleado

### Tiempos de desarrollo

## Prioridades (explicadas en puntos anteriores)

Simplicidad

Compatibilidad

Sencillez -

## Dificultades

Interacción muy compleja de ajustar.

## Detalle de las funcionalidades

Importar vídeo

Importar imagen

Determinar duración imagen

Determinar rango vídeo

Independencia de formatos

Adición de notas con formato

Adición de stickers

WSIWYG multinavegador: javascript css dropzone jquery bootstrap vanilla js

Seguridad de sesiones

Cookies

## Futuros desarrollos

HTTPS

Captura webcam

Transiciones.

## Pruebas

Unitarias

Sistema

Distintos sistemas operativos

Distintos navegadores

Distintas resoluciones d epantalla.

Sin conexión a internet (en caso de querer aplicación stand-alone).

Escape de caracteres

## Prestaciones

Necesidades soft y hard

Recompilación ffmpeg para utilizar CUDA

# Análisis y presentación de resultados, con los principales hallazgos.

Incluso los programas y formatos más reconocidos pueden estar sujetos a una revisión --> nuevas necesidades, clientes con mayor capacidad y menor paciencia, objetivos más claros, no siempre existe un deseo de exploración de aplicaciones.

“El software como persona” -> conocer el software – expertise-

# Conclusiones más relevantes y aportaciones teóricas y metodológicas.

# Anexo Entrevistas

# Bibliografía

Babich, N. (18 de octubre de 2019). *User Centered Design Principles & Methods.* Obtenido de xd.adobe.com: https://xd.adobe.com/ideas/principles/human-computer-interaction/user-centered-design/

Cooper, A. (2014). *About Face.* Editorial John Wiley & Sons, Inc.

Donnelly, J. (4 de noviembre de 2021). *The Evolution of Video Editing: From the Moviola to Machine Learning.* Obtenido de massive.io: https://massive.io/blog/the-evolution-of-video-editing/

Harris, M. (2014). *The End of Absence.* Editorial Portfolio Penguin.

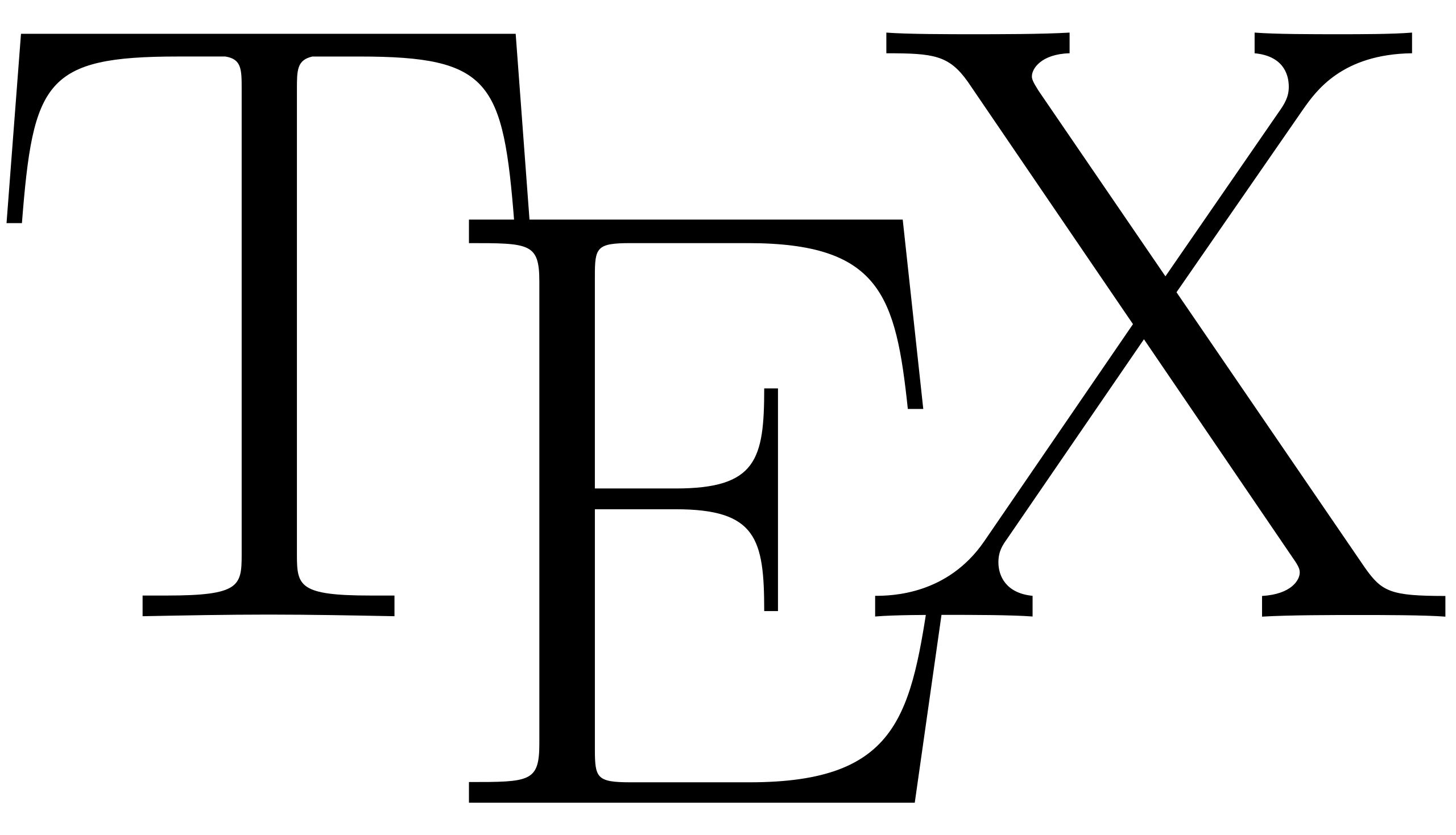
IDEO.org. (2015). *The Field Guide to Human-Centered Design.* Authorhouse.

Interaction Design Foundation. (s.f.). *Design Thinking.* Recuperado el 9 de enero de 2022, de Interaction Design Foundation: https://www.interaction-design.org/literature/topics/design-thinking

Nielsen, L. (2012). *Personas - User Focused Design.* Editorial Springer.

Salazar, K. (28 de enero de 2018). *Why Personas Fail.* Obtenido de Nielsen Norman Group: https://www.nngroup.com/articles/why-personas-fail/

Watson, I. (6 de septiembre de 2018). *Unmasking uninstalls: Three data points to think about .* Obtenido de adjust.com: https://www.adjust.com/blog/unmasking-uninstalls/

1. <https://github.com/jacnc17/TFM> [↑](#footnote-ref-1)
2. <https://spring.io/projects/spring-boot> [↑](#footnote-ref-2)
3. <https://www.thymeleaf.org/> [↑](#footnote-ref-3)
4. <https://ffmpeg.org/> [↑](#footnote-ref-4)
5. <https://blog.youtube/press/> [↑](#footnote-ref-5)
6. <https://www.statista.com/statistics/266587/percentage-of-internet-users-by-age-groups-in-the-us/> [↑](#footnote-ref-6)
7. El primer “corte” en una película se considera que se pudo ver en “*Come along, do!*”. en 1898 (http://www.screenonline.org.uk/film/id/444430/). [↑](#footnote-ref-7)
8. <https://www.adobe.com/creativecloud/video/hub/ideas/what-is-non-linear-editing> [↑](#footnote-ref-8)
9. DaVinci Resolve 17, en <https://www.blackmagicdesign.com/es/products/davinciresolve/studio> [↑](#footnote-ref-9)
10. <https://shotcut.org> [↑](#footnote-ref-10)
11. <https://www.openshot.org/> [↑](#footnote-ref-11)
12. <https://lwks.com/> [↑](#footnote-ref-12)
13. <https://clipchamp.com/en/> [↑](#footnote-ref-13)
14. <https://www.kapwing.com> [↑](#footnote-ref-14)
15. <https://www.powtoon.com/> [↑](#footnote-ref-15)
16. Estilización del nombre en tipografía TeX (). [↑](#footnote-ref-16)
17. Pese a que el término “persona” debe traducirse a español como “personaje” o “arquetipo”, se utilizará en este documento la nomenclatura original en inglés debido a su condición de estándar. [↑](#footnote-ref-17)
18. Foto de [Annie Spratt](https://unsplash.com/@anniespratt?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) vía [Unsplash](https://unsplash.com/s/photos/child-computer?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) [↑](#footnote-ref-18)
19. Foto de [nappy](https://www.pexels.com/es-es/@nappy?utm_content=attributionCopyText&utm_medium=referral&utm_source=pexels) vía [Pexels](https://www.pexels.com/es-es/foto/foto-de-hombre-con-bebe-3536643/?utm_content=attributionCopyText&utm_medium=referral&utm_source=pexels) [↑](#footnote-ref-19)
20. Foto de [nappy](https://www.pexels.com/es-es/@nappy?utm_content=attributionCopyText&utm_medium=referral&utm_source=pexels) vía [Pexels](https://www.pexels.com/es-es/foto/foto-de-hombre-con-bebe-3536643/?utm_content=attributionCopyText&utm_medium=referral&utm_source=pexels) [↑](#footnote-ref-20)
21. <https://ffmpeg.org/ffmpeg-utils.html#Video-size> [↑](#footnote-ref-21)
22. El director de fotografía Vittorio Storaro, ASC, propuso el formato Univisium con esta proporción de aspecto, con el objetivo de aprovechar al máximo el espacio en celuloide de 35 o 65mm. [↑](#footnote-ref-22)
23. <https://ads.tiktok.com/help/article?aid=10002742> [↑](#footnote-ref-23)
24. <https://ffmpeg.org/ffmpeg-codecs.html#Video-Decoders> [↑](#footnote-ref-24)
25. <https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs> [↑](#footnote-ref-25)