

Universidad de Valladolid

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE ESCRITORIO DEDICADA A LA COMPOSICIÓN FOTOGRAFICA

Autor:

D. Pablo Martín Junquera

Tutora:

Dra. Dña. Miriam Antón Rodríguez

Valladolid, 4 de Junio de 2015



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica



TRABAJO FIN DE GRADO

TÍTULO: Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica: Autodesign

AUTOR: D. Pablo Martín Junquera

TUTORA: Dña. Míriam Antón Rodríguez

DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería
Telemática

TRIBUNAL

PRESIDENTA: Dña. M^a Ángeles Pérez Juárez

VOCAL: Dña. Míriam Antón Rodríguez

SECRETARIO: D. David González Ortega

SUPLENTE 1: D. Mario Martínez Zarzuela

SUPLENTE 2: D. Francisco Javier Díaz Pernas

FECHA: Julio de 2015

CALIFICACIÓN:



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica



AGRADECIMIENTOS

“A mis padres, por apoyarme y confiar en mí hasta el final; a mis hermanos, por animarme en todo momento; a Claudia, por su ayuda, apoyo y ánimos en los buenos y malos momentos; a mi tutora, Miriam, por toda su ayuda, esfuerzo y dedicación”



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica



RESUMEN

En este Trabajo Fin de Grado se propone el diseño y desarrollo de una aplicación software dedicada a la edición digital de imágenes. El programa ofrecerá las capacidades y medios necesarios para realizar composiciones fotográficas a partir de diferentes archivos de imagen y tendrá como finalidad obtener una imagen que resulte de la correcta combinación de éstas.

El objetivo es que los usuarios puedan realizar una composición fotográfica de forma básica e intuitiva que después será procesada de forma más compleja mediante un programa externo. La aplicación tendrá un enfoque a la edición digital de imagen desde los niveles más bajos, permitiendo a los usuarios más inexpertos expresar su creatividad e ideas en este campo. Por lo tanto, un aspecto clave será la experiencia de usuario basada en la sencillez y facilidad de uso de la propia herramienta.

El trabajo a realizar comprende todas las fases en el desarrollo de un proyecto software de estas características, abarcando desde el extremo más teórico hasta el más práctico, y tanto en el ámbito más técnico como en el entorno de un usuario final. Para alcanzar los objetivos se hace uso de un amplio abanico de tecnologías, entre las que destacan los lenguajes de programación MySQL y Python, junto con las librerías convenientes, y el programa Eclipse como principal herramienta de desarrollo.

PALABRAS CLAVE

Desarrollo software, edición digital de imagen, Python, Eclipse, MySQL.



ABSTRACT

In this final project the design and development of a software application dedicated to the digital edition of images is proposed. The application will offer the abilities and means needed to create photographic compositions from different image files and will be destined to obtain a resulting image from the combination of these.

Its goal is for the users to be able to carry out a photographic composition in a basic and intuitive way which will be processed in a more complex way by an external program afterwards. The application will be aimed at the digital edition of an image at its lower levels allowing the most inexperienced users to express their creativity and ideas in this field. Therefore, a key aspect of this application will be the user's experience based in the simplicity and the ease in the use of this tool.

The work to do includes every phase in the development of a software project of these characteristics, covering from the most theoretical aspects to the most practical ones, including the most technical issue as well as the final user environment. To achieve the goals, a wide variety of technologies have been used, among which the programming languages MySQL and Python stand out, along with the convenient libraries, and the application Eclipse as the main developing tool.

KEYWORDS

Software development, digital image edition, Python, Eclipse, MySQL



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica



ÍNDICE

Trabajo Fin de Grado	3
Agradecimientos	5
Resumen	7
Abstract.....	8
Capítulo 1: Introducción	22
1. Introducción al Proyecto	22
2. Contexto y Motivación	23
3. Objetivos.....	24
4. Fases y Metodología	25
5. Introducción al Documento.....	27
Capítulo 2: Conocimientos Teóricos	30
1. Conceptos de Software	30
1.1. Tipos de aplicaciones.....	30
1.2. Sistemas operativos	35
2. Conceptos de imagen	36
2.1. Tipos de imagen digital	36
2.2. Formatos de imagen.....	38
2.3. Canales de color. El canal Alfa	45
3. Tecnologías	49
3.1. Lenguajes de programación: Python.....	49
3.2. Librerías de Python.....	53
3.3. Herramientas de desarrollo: Eclipse	55
3.4. Servidor: XAMPP	58
3.5. Bases de datos.....	62
3.6. Arquitecturas software para el intercambio de información: REST.....	66
3.7. Formatos de intercambio de información: JSON	68



Capítulo 3: Desarrollo del Proyecto 71

1. Archivos Base.....	71
2. Estructura de Ficheros y Directorios	72
3. Ficheros de Código	74
3.1. __init.....	74
3.2. main.py.....	74
3.3. panel_usuario.py.....	75
3.4. panel_imagenes.py.....	75
3.5. panel_edicion.py	75
3.6. panel_proyecto.py	75
3.7. imagen.py.....	76
3.8. bd.py.....	76
3.9. config.py	76
3.10. setup.py.....	76
4. Interfaz Gráfica de Usuario (GUI)	77
4.1. Ventana de inicio.....	77
4.2. Ventana de registro.....	78
4.3. Ventana principal	78
4.4. Otras ventanas	83
5. Base de Datos	83
5.1. Tabla usuarios	83
5.2. Tablas de escenarios, personajes y complementos	85
6. Funcionalidades.....	86
6.1. Registrarse.....	86
6.2. Entrar.....	87
6.3. Modificar los datos de usuario	88
6.4. Cambiar la clave de usuario	89
6.5. Enviar un correo	90
6.6. Acceder al área de administración.....	91
6.7. Editar las preferencias.....	91
6.8. Mostrar información del programa.....	92



6.9.	Abrir el manual de usuario	93
6.10.	Buscar escenarios	93
6.11.	Mostrar información de escenario	94
6.12.	Cargar escenario	94
6.13.	Buscar personajes	95
6.14.	Cargar personaje	96
6.15.	Buscar complementos	97
6.16.	Cargar complemento	97
6.17.	Cargar imagen externa	98
6.18.	Añadir elemento a la composición	99
6.19.	Modificar posición	99
6.20.	Modificar tamaño	100
6.21.	Cambiar de plano	101
6.22.	Voltear horizontalmente	101
6.23.	Eliminar elemento de la composición	102
6.24.	Editar un efecto	103
6.25.	Aplicar un filtro	103
6.26.	Crear nuevo proyecto	104
6.27.	Guardar proyecto	105
6.28.	Abrir Proyecto	105
6.29.	Exportar composición	106
6.30.	Abrir proyecto de usuario	107
6.31.	Cambiar de usuario	108
6.32.	Salir de la aplicación	109
7.	Conexión Aplicación – Servidor	110
8.	Características	112
8.1.	Responsividad	112
8.2.	Usabilidad	114
9.	Paquete de Instalación	115
9.1.	Archivo ejecutable	116
9.2.	Instalador	116



Capítulo 4: Manual de Usuario 119

- 1. Guía de Instalación 119
- 2. Usuario..... 120
 - 2.1. Registro 120
 - 2.2. Entrar..... 120
 - 2.3. Modificar tus datos 121
- 3. Composición 121
 - 3.1. Escenarios..... 122
 - 3.2. Personajes 122
 - 3.3. Complementos 125
 - 3.4. Imágenes propias 128
 - 3.5. Composición 130
- 4. Proyecto..... 133
- 5. Otros 135

Capítulo 5: Análisis Final 139

- 1. Presupuesto Económico 139
- 2. Conclusiones..... 141
- 3. Líneas Futuras 144
 - 3.1. Mejoras de carácter general 144
 - 3.2. Solución de errores 145
 - 3.3. Mejoras de usabilidad 146
 - 3.4. Mejoras en las funcionalidades existentes 146
 - 3.5. Incorporación de nuevas funcionalidades 147

Bibliografía 150

Anexos..... 154

- 1. Instalación del Lenguaje de programación: Python 154
- 2. Instalación y Configuración del Entorno de Desarrollo: Eclipse + PyDev..... 158
- 3. Incorporación de Librerías Externas 166
- 4. Configuración del Servidor Remoto 167



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica



ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1 – Arquitectura web: cliente-servidor- base de datos.....</i>	<i>32</i>
<i>Ilustración 2 – Imagen vectorial / imagen en mapa de bits.....</i>	<i>37</i>
<i>Ilustración 3 – Compresión con pérdida JPEG.....</i>	<i>39</i>
<i>Ilustración 4 – Formatos básicos de imagen digital.....</i>	<i>41</i>
<i>Ilustración 5 – Formatos complejos de imagen digital.....</i>	<i>42</i>
<i>Ilustración 6 – Plano sin transparencia.....</i>	<i>44</i>
<i>Ilustración 7 – Plano con transparencia.....</i>	<i>44</i>
<i>Ilustración 8 – Canales RGB + canal Alfa = RGBA.....</i>	<i>45</i>
<i>Ilustración 9 – Eclipse + PyDev + wxPython.....</i>	<i>57</i>
<i>Ilustración 10 – Servidor en red.....</i>	<i>58</i>
<i>Ilustración 11 – Panel de control de un servidor XAMPP.....</i>	<i>61</i>
<i>Ilustración 12 – Ejemplo de directorio de escenario.....</i>	<i>73</i>
<i>Ilustración 13 – Vista de la ventana de registro.....</i>	<i>78</i>
<i>Ilustración 14 – Vista de la ventana de inicio.....</i>	<i>78</i>
<i>Ilustración 15 - Vista de la ventana de bienvenida.....</i>	<i>78</i>
<i>Ilustración 16 – Vista de la ventana principal.....</i>	<i>79</i>
<i>Ilustración 17 – Vista de la barra de herramientas.....</i>	<i>79</i>
<i>Ilustración 18 – Vista del panel de usuario.....</i>	<i>80</i>
<i>Ilustración 19 – Vista del panel de imágenes.....</i>	<i>80</i>
<i>Ilustración 20 – Vistas del panel de edición.....</i>	<i>81</i>
<i>Ilustración 21 – Vista de la ventana de exportación.....</i>	<i>82</i>
<i>Ilustración 22 – Vista de la ventana de enviar correo.....</i>	<i>82</i>
<i>Ilustración 23 – Vista de la ventana de preferencias.....</i>	<i>82</i>
<i>Ilustración 24 – Vistas de ventanas informativas y de control.....</i>	<i>82</i>
<i>Ilustración 25 – Tabla usuarios.....</i>	<i>84</i>
<i>Ilustración 26 – Tabla personajes.....</i>	<i>85</i>
<i>Ilustración 27 – Tabla complementos.....</i>	<i>85</i>
<i>Ilustración 28 – Tabla escenarios.....</i>	<i>85</i>
<i>Ilustración 29 – Conexión aplicación - servidor MySQL.....</i>	<i>111</i>
<i>Ilustración 30- Responsividad de la aplicación.....</i>	<i>113</i>
<i>Ilustración 31 – Instalador.....</i>	<i>117</i>
<i>Ilustración 32 – Comando “python”.....</i>	<i>156</i>
<i>Ilustración 33 – Comando “python” no reconocido.....</i>	<i>156</i>
<i>Ilustración 34 – Configuración de variables de entorno.....</i>	<i>157</i>
<i>Ilustración 35 – Variable de entorno “Path”.....</i>	<i>158</i>
<i>Ilustración 36 – Ventana de “Nuevo software” de Eclipse.....</i>	<i>160</i>
<i>Ilustración 37 – Ventana “Añadir repositorio” de Eclipse.....</i>	<i>160</i>
<i>Ilustración 38 – Ventana de instalación de PyDev en Eclipse.....</i>	<i>161</i>
<i>Ilustración 39 - Ventana de instalación de PyDev en Eclipse (2).....</i>	<i>161</i>



<i>Ilustración 40 – Ventana de instalación de PyDev en Eclipse (3)</i>	162
<i>Ilustración 41 – Ventana de instalación de PyDev en Eclipse (3)</i>	163
<i>Ilustración 42 – Ventana de instalación de PyDev en Eclipse (4)</i>	163
<i>Ilustración 44 – Ventana de instalación de PyDev en Eclipse (6)</i>	164
<i>Ilustración 43 – Ventana de instalación de PyDev en Eclipse (5)</i>	164
<i>Ilustración 45 – Ventana de preferencias de Eclipse</i>	165
<i>Ilustración 46 – Agregar librerías en Eclipse</i>	167
<i>Ilustración 47 – Panel de control XAMPP con Apache y MySQL</i>	168
<i>Ilustración 48 – Esquema de redirección de puertos</i>	171



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica



ÍNDICE DE GRÁFICOS Y TABLAS

<i>Gráfico 1 – Índices de popularidad de los lenguajes de programación 2004-2014</i>	52
<i>Gráfico 2 – Oferta de empleo según los lenguajes de programación</i>	52
<i>Gráfico 3 – Distribución de protocolos en APIs</i>	67
<i>Gráfico 4 – Comparativa de características SOAP y REST</i>	67
<i>Tabla 1– Comparativa de formatos de imagen</i>	43
<i>Tabla 2 – Índices de popularidad de los lenguajes de programación</i>	52
<i>Tabla 3 – Tipos de servidores</i>	59
<i>Tabla 4 – Caso de uso: Registrarse</i>	87
<i>Tabla 5 – Caso de uso: Entrar</i>	88
<i>Tabla 6 – Caso de uso: Modificar los datos de usuario</i>	89
<i>Tabla 7 – Caso de uso: Cambiar clave de usuario</i>	90
<i>Tabla 8 – Caso de uso: Enviar un correo</i>	90
<i>Tabla 9 – Caso de uso: Acceder al área de administración</i>	91
<i>Tabla 10 – Caso de uso: Editar las preferencias</i>	92
<i>Tabla 11 – Caso de uso: Mostrar información del programa</i>	92
<i>Tabla 12 – Caso de uso: Abrir el manual de usuario</i>	93
<i>Tabla 13 – Caso de uso: Buscar escenarios</i>	94
<i>Tabla 14 – Caso de uso: Mostrar información de escenario</i>	94
<i>Tabla 15 – Caso de uso: Cargar escenario</i>	95
<i>Tabla 16 – Caso de uso: Buscar personajes</i>	96
<i>Tabla 17 – Caso de uso: Cargar personaje</i>	96
<i>Tabla 18 – Caso de uso: Buscar complementos</i>	97
<i>Tabla 19 – Caso de uso: Cargar complemento</i>	98
<i>Tabla 20 – Caso de uso: Cargar imagen externa</i>	98
<i>Tabla 21 – Caso de uso: Añadir elemento a la composición</i>	99
<i>Tabla 22 – Caso de uso: Modificar posición</i>	100
<i>Tabla 23 – Caso de uso: Modificar tamaño</i>	100
<i>Tabla 24 – Caso de uso: Cambiar de plano</i>	101
<i>Tabla 25 – Caso de uso: Voltrear horizontalmente</i>	102
<i>Tabla 26 – Caso de uso: Eliminar elemento de la composición</i>	102
<i>Tabla 27 – Caso de uso: Editar un efecto</i>	103
<i>Tabla 28 – Caso de uso: Aplicar un filtro</i>	104
<i>Tabla 29 – Caso de uso: Crear nuevo proyecto</i>	104
<i>Tabla 30 – Caso de uso: Guardar proyecto</i>	105



<i>Tabla 31 – Caso de uso: Abrir proyecto</i>	<i>106</i>
<i>Tabla 32 – Caso de uso: Exportar composición</i>	<i>107</i>
<i>Tabla 33 – Caso de uso: Abrir proyecto de usuario</i>	<i>108</i>
<i>Tabla 34 – Caso de uso: Cambiar de usuario</i>	<i>109</i>
<i>Tabla 35 – Caso de uso: Salir de la aplicación</i>	<i>109</i>



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica



CAPÍTULO 1: INTRODUCCIÓN

1. INTRODUCCIÓN AL PROYECTO

La edición digital de imagen es una idea originada durante el periodo de revolución digital acontecido durante las últimas décadas, y como tal, ha sufrido una fuerte evolución. La modificación de imágenes en formato digital era un concepto inalcanzable hace unos años, mientras que en la actualidad las posibilidades disponibles en este campo son extraordinarias e innumerables.

El software y el hardware informáticos son los principales factores que han influido en el impulso que ha experimentado la edición de imagen en los últimos años. Estos ámbitos, software y hardware, son los que realmente han sufrido un crecimiento exponencial durante la mencionada revolución digital, permitiendo el desarrollo de la edición digital hasta el punto en el que se encuentra actualmente. No debemos obviar tampoco la base de la imagen, la fotografía, la cual también ha sido afectada en gran medida por estos mismos factores.

En nuestra sociedad, día a día tenemos contacto con imágenes digitales, ya que prácticamente vivimos “enganchados” y somos dependientes de los dispositivos digitales, los cuales permanentemente hacen uso de ellas. Tal es su importancia, que es difícil encontrar imágenes, en cualquier ámbito tecnológico, que no hayan sufrido un proceso de adaptación o mejora previo a su uso. La edición digital de imagen es la encargada de llevar a cabo estas transformaciones gráficas. Ésta tiene influencia desde el ámbito laboral o profesional hasta el perfil más aficionado o incluso artístico. Multitud de sectores laborales hacen uso del procesado de imagen digital, a pesar de no estar directamente enfocados a este tipo de edición, utilizándolo como medio para obtener sus objetivos. Sin embargo, otros sectores están enfocados a la imagen y a su edición como propia finalidad. Atendiendo a los sectores aficionado y artístico de la fotografía e imagen, la edición digital adquiere también una gran importancia entre sus simpatizantes.

En el proyecto realizado durante el presente trabajo se aborda la edición digital de imagen desde el punto de vista del software y desde el extremo más aficionado, proponiendo una solución alternativa a las existentes, aunque con unas características y un enfoque bien diferenciados.



2. CONTEXTO Y MOTIVACIÓN

El Grado en Ingeniería de Tecnologías de Telecomunicación abarca un extenso campo de trabajo, lo que permite a su finalización un amplio abanico de posibilidades en las que especializarse. En mi caso personal, los campos que más interés me suscitan son, por un lado, el desarrollo software, y por otro lado el ámbito tanto de sonido como de imagen. El primer área mencionada, referente al desarrollo de software, se ha visto fomentado por la realización de las prácticas en empresa, y a raíz de esta situación nace la idea de unir parcialmente ambos campos de trabajo en este proyecto. La idea originada consiste en desarrollar una aplicación software dedicada a la edición digital de imagen, que permita acercar a los usuarios al mundo de la edición fotográfica desde los niveles más bajos.

Se plantea este trabajo como respuesta a la solicitud de una empresa ficticia especializada en fotografía y edición de imagen de un software que facilite y agilice sus tareas de procesado/edición. Tomaré el rol de desarrollador o programador de la empresa de desarrollo software que llevará a cabo dicho producto solicitado.

La empresa cliente está especializada en la composición y edición fotográfica, cuyo objetivo es ofrecer ciertos servicios a partir de estas capacidades. El objetivo final de la aplicación propuesta por dicha empresa consiste en ofrecer a sus respectivos clientes la posibilidad de diseñar sus propias composiciones de una forma sencilla, ágil e intuitiva. La supuesta empresa se dedica a realizar estas composiciones artísticas de forma profesional mediante potentes programas de edición de imagen, mientras que los clientes solicitan las composiciones personalizadas sobre su familia, amigos o cualquier grupo de personas enviando sus fotografías. La aplicación Autodesign solicitada permitiría a los clientes crear y transmitir su idea sobre la composición final a la empresa de profesionales. De esta forma, un cliente tiene la posibilidad de diseñar de una forma muy básica, sencilla e intuitiva su propio producto final, que después será elaborado de forma compleja y precisa por los profesionales de la empresa. Los retratos fotográficos elaborados tendrán un enfoque cómico característico introducido por las propias imágenes ofrecidas por la empresa. Estas imágenes son un conjunto de posibles escenarios y figuras o personajes que el cliente podrá elegir a su gusto para realizar cada una de las composiciones artísticas.

A raíz de la idea propuesta y las metas conceptuales del proyecto, es posible fijar también los objetivos a nivel técnico del producto que se va a diseñar.



3. OBJETIVOS

Se plantea por tanto el desarrollo e implementación de una solución software capaz de acercar la edición digital de imagen a un público aficionado y principiante. Por lo tanto, se enfocará desde los niveles básicos en la modificación de imágenes y teniendo como puntos clave el diseño centrado en el usuario y la sencillez.

La aplicación desarrollada durante este proyecto consistirá en una primera versión de la aplicación final (denominada “Autodesign”) y, por lo tanto, deberá asentar las bases de ésta cumpliendo la mayor cantidad de objetivos en la medida de lo posible. A priori, existe cierta incertidumbre acerca de la complejidad que suponen algunos de los objetivos y en consecuencia no se trabajará bajo una idea cerrada. Ciertas finalidades de carácter secundario del software pueden verse modificadas durante la fase de desarrollo, mientras que los objetivos más básicos permanecerán inalterados.

Se establece como fin principal y básico de la aplicación desarrollada, “Autodesign 1.0”, la posibilidad de llevar a cabo la edición digital de imágenes mediante la descomposición de imágenes en múltiples capas o planos. Se realizarán ediciones independientes en los distintos planos por separado, permitiendo además ser antepuestos o postergados en relación a otros planos. Las composiciones realizadas en la aplicación estarán formadas a partir de una serie de escenarios, personajes y complementos disponibles en la aplicación, que combinados entre sí generarán una imagen final. Se propone alternativamente ofrecer la posibilidad de introducir imágenes externas a la aplicación, propias del usuario; así como la posibilidad de editarlas de forma básica y/o combinarlas con las propias imágenes incluidas en el programa.

Conseguir un correcto diseño gráfico de la aplicación será otro de los puntos fuertes. Dotando a la interfaz gráfica de un aspecto visiblemente atractivo y, ante todo, capaz de ofrecer características notables de usabilidad. Para ello el diseño estará basado en la simplicidad y la organización de los componentes gráficos, enfocados a conseguir la mejor experiencia de usuario posible. Recordemos que el control de la aplicación ha de ser intuitivo y sencillo, puesto que debe ser accesible para todo el público. Se valorará también la posibilidad de incorporar responsividad a esta interfaz de usuario.

Se investigará la posibilidad de incorporar finalidades secundarias y se desarrollarán aquellas que se ajusten en mayor medida al nivel de complejidad de la aplicación en cuanto a programación se refiere. Se mencionan, entre otras, la capacidad de la



CAPÍTULO 1: INTRODUCCIÓN

aplicación para llevar una gestión de proyectos y/o pedidos de cada usuario, el desarrollo de un área de administración dedicado principalmente a gestionar algunos recursos, la importación y exportación de imágenes, la incorporación de funciones básicas de edición de imagen o el proceso de compra y pedidos.

Han sido planteados gran cantidad de objetivos alternativos, algunos aparentemente complejos. Y, como ya se ha mencionado anteriormente, se implementarán según se crea conveniente durante el proceso de desarrollo. Incluso en el aspecto de funcionalidades del programa, se podrán implementar a mayores todas a aquellas que se considere que puedan mejorar la aplicación.

A nivel personal, y dejando la aplicación a un lado, se pretenden adquirir nuevos conocimientos a niveles teórico y práctico, y obtener los beneficios que supone una experiencia de estas características. Estos objetivos son:

- i) Poner en práctica las capacidades obtenidas durante el periodo de realización del Grado en Ingeniería de Tecnologías de Telecomunicación.
- ii) Utilizar correctamente los conceptos teóricos obtenidos durante el Grado y complementarlos mediante técnicas de autoaprendizaje.
- iii) Adquirir la experiencia en el desarrollo de un proyecto de estas dimensiones. Mejorar para ello en labores organizativas, gestión de recursos y planificación.
- iv) Mejorar las metodologías y conceptos de programación, aprendiendo si es necesario un nuevo lenguaje de programación.
- v) Ampliar los conocimientos acerca de las nuevas tecnologías en el ámbito laboral y sobre los temas abordados en el trabajo: edición digital de imagen y desarrollo software.

4. FASES Y METODOLOGÍA

Antes de entrar en detalle en los siguientes aspectos, se hace necesario contextualizar y analizar la propuesta del proyecto que se realiza. Es imprescindible comprender la idea principal de éste, para fijar posteriormente en más detalle los objetivos más concretos. En el caso de este trabajo, la idea original viene por parte de la empresa de fotografía que se mencionaba anteriormente, por lo que se hace necesaria una fase previa de comunicación con el fin de obtener toda la información necesaria sobre las funcionalidades y requisitos que se esperan del programa software a realizar.



CAPÍTULO 1: INTRODUCCIÓN

En segundo lugar, se realiza una etapa de aprendizaje de conceptos teóricos, los cuales formarán los cimientos del propio proyecto. Es importante adquirir los conocimientos necesarios sobre los temas involucrados en la elaboración del proyecto mediante un proceso de documentación. Estos conceptos fundamentales permitirán realizar un correcto análisis en esta primera fase, permitiendo tomar decisiones coherentes y acertadas durante todo el proceso de desarrollo. Optar por el camino correcto desde un primer momento puede suponer un importante ahorro en términos de tiempo y esfuerzo.

Una vez está claro cuál es el proyecto al que nos enfrentamos es conveniente llevar a cabo una correcta planificación y organización. En términos temporales, es importante establecer fechas límite para la finalización de las distintas etapas y la consecución de los objetivos, al menos de forma aproximada. Pero también es primordial la organización de estas etapas y objetivos en la línea temporal disponible. Estructurar correctamente el diseño y desarrollo de la aplicación, fraccionando estos según objetivos, complejidad y prioridades, favorece notablemente estos procesos. En definitiva, ésta es una etapa clave que puede afectar de forma muy positiva o muy negativa a las fases posteriores y, por lo tanto, a la globalidad del trabajo.

Antes de comenzar con el desarrollo propiamente dicho del proyecto, es importante adquirir los recursos necesarios para la elaboración del mismo. A modo de recordatorio, en el escenario de este trabajo, los recursos mencionados hacen referencia a la edición de imagen y al desarrollo software. Se hace imprescindible conseguir y configurar el entorno de desarrollo. En este caso y en líneas generales, formado por un ordenador, acceso a Internet, el lenguaje de programación Python y algunas de sus librerías externas o de terceros, el software Eclipse junto con el plugin PyDev, y el servidor XAMPP. En el ámbito de la imagen, se recopilan y adaptan las imágenes a los requisitos impuestos por la siguiente etapa.

Con los objetivos cumplidos en las fases previas, es posible afrontar con ciertas garantías la fase de desarrollo del proyecto. Consiste en dar forma a éste haciendo uso de los conocimientos y recursos obtenidos, siguiendo la planificación establecida, y sin perder de vista la idea y objetivos establecidos en la propuesta del proyecto. En el presente trabajo de desarrollo software, la programación en Python es la principal herramienta de desarrollo, aunque no la única. Es importante destacar que durante esta fase se lleva cabo, de forma paralela, un proceso de aprendizaje, ya que se desconocía el lenguaje de programación elegido. En un primer momento se adquieren los conceptos teóricos básicos mediante documentación web en forma de tutoriales.



CAPÍTULO 1: INTRODUCCIÓN

La metodología seguida para la elaboración de gran parte del código que da forma a la aplicación se puede sintetizar en los siguientes pasos:

- i) Planteamiento de la idea: Se fija inicialmente un objetivo a realizar.
- ii) Documentación vía web sobre la resolución del problema planteado, analizando además sus posibles soluciones y su viabilidad. Principalmente se ha utilizado documentación oficial y foros dedicados a la programación en Python.
- iii) Desarrollo del código en base a la información recogida.
- iv) Pruebas sobre el nuevo código introducido. En caso de fallo o error, primero se localiza y finalmente se analiza el propio fallo.
- v) Segunda fase de documentación y búsqueda de información en torno al fallo analizado.
- vi) Resolución del error de la mejor forma posible o adecuada para cada situación.

Los objetivos planteados durante la fase de planificación son adaptados según las limitaciones impuestas por la complejidad y viabilidad de las ideas. Los objetivos más complejos serán pospuestos en la planificación a momentos de mayor dominio del lenguaje, o incluso a las posteriores versiones de la aplicación. Esto no debe tomarse como un aspecto negativo, ya que es una cuestión organizativa lógica. A mayor dominio del lenguaje y herramientas, mejores son las soluciones elaboradas, y mayores las posibilidades de éxito en la finalización de un objetivo.

A continuación, se realiza una exhaustiva fase de pruebas que permitan comprobar el correcto funcionamiento del proyecto realizado. La solución de cada problema localizado en esta etapa permite perfeccionar el resultado final. Se realiza un análisis a niveles técnico y funcional de la aplicación, comprobando y perfeccionando la totalidad de funcionalidades. De igual manera se efectúan pruebas a nivel de usuario con el fin de pulir tanto la estética de la interfaz de usuario como las características de usabilidad de la aplicación.

Finalmente, se fijarán las bases de la próxima versión del proyecto, en función de un estudio de los resultados obtenidos, las carencias o puntos débiles de la aplicación y los objetivos finalizados con éxito.

5. INTRODUCCIÓN AL DOCUMENTO

El presente documento está estructurado en cinco capítulos principales, cuyos contenidos están organizados en los distintos apartados de los que consta cada capítulo. Finalmente se incluyen los capítulos “Bibliografía” y “Anexos” para completar la información.



CAPÍTULO 1: INTRODUCCIÓN

Capítulo 1: Introducción.

Esta sección está dedicada al planteamiento del proyecto que se desea realizar, describiendo la temática que rodea el proyecto, sus puntos clave y las finalidades que pretende conseguir. Se describen también la metodología y estructura del propio trabajo.

Capítulo 2: Conocimientos teóricos.

Se desarrollan los conceptos previos necesarios tanto para elaborar el proyecto como para comprender los posteriores conceptos del trabajo realizado. Para ello, se estudian aspectos clave relacionados con la temática del trabajo: aplicaciones, software, imagen, tecnología y programación.

Capítulo 3: Desarrollo del proyecto.

En este capítulo se abordan todos los aspectos relacionados con la elaboración de la aplicación. Se detallan desde los aspectos más técnicos, relacionados con los ficheros de código y la base de datos, hasta los más cercanos a un usuario, como la interfaz gráfica o las propias características funcionales y visuales de la aplicación.

Capítulo 4: Manual de usuario.

Recoge de forma ordenada las pautas y metodologías para el correcto uso de la aplicación por parte de los usuarios. Se detallan las funcionalidades de la aplicación a nivel de usuario.

Capítulo 5: Análisis final

Se incluyen análisis y resultados realizados tras el desarrollo del proyecto. Contiene un presupuesto económico orientativo, descripción de la proyección de futuro de la aplicación y conclusiones extraídas a través del trabajo realizado.

Bibliografía

Recoge las fuentes de las que se ha obtenido información útil para el desarrollo del trabajo.

Anexos

Se incluyen apartados técnicos que amplían la información relacionada con la configuración del entorno de trabajo para proyectos de características similares.



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica

CAPÍTULO 1: INTRODUCCIÓN



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

1. CONCEPTOS DE SOFTWARE

1.1. Tipos de aplicaciones

Las aplicaciones son programas que se utilizan como herramienta para llevar a cabo el desarrollo de una tarea específica en una computadora. A la hora de plantearnos el desarrollo de una aplicación el primer dilema que se nos plantea es el tipo de aplicación que queremos realizar. Las opciones más habituales son las aplicaciones software de escritorio, las aplicaciones WEB y las aplicaciones móviles. A continuación se detallan estos conceptos, sus ventajas e inconvenientes en forma de comparativa.

Aplicaciones móviles: Son aplicaciones informáticas desarrolladas con el propósito de ser utilizadas en dispositivos portátiles tales como teléfonos móviles, tabletas... En el desarrollo de este tipo de software hay que tener muy en cuenta las limitaciones que cada dispositivo móvil impone.

Aplicaciones WEB: Son aquellas aplicaciones que se pueden ejecutar a través de un navegador WEB. Su principal característica es que son ejecutadas en un servidor, y es éste el que devuelve un código legible por un navegador de Internet en forma de programa que el usuario puede manejar.

Aplicaciones de escritorio: Este tipo de aplicaciones se instalan en la propia computadora según es el sistema operativo de ésta. Trabajan, es decir, se ejecutan de forma local en lo que se conoce como “lado del cliente”, aunque se pueden conectar de forma remota a un servidor. Es por esto que también se las conoce como aplicaciones cliente-servidor.

El objetivo de este trabajo está enfocado a realizar una aplicación que se ejecute en una computadora o PC, es decir, un dispositivo no móvil, por lo tanto la comparación se lleva a cabo descartando las aplicaciones móviles y analizando las características que nos ofrecen las otras soluciones. Las ventajas de una solución se corresponden con los inconvenientes de la otra, pues son opciones bastante opuestas.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Ventajas de las aplicaciones WEB [Moreno, J.M. (2010)]:

- i. **No requiere instalar software especial:** Para acceder a un software web sólo necesitamos disponer de un navegador de páginas web (Internet Explorer, Firefox, Opera, Chrome, etc), los cuales suelen venir con el propio sistema operativo. No es necesario tener nada más. Debido a la arquitectura de las aplicaciones web, el navegador suele quedar relegado a mostrar el interfaz de usuario (menús, opciones, formularios, etc), mientras que toda la compleja lógica de negocio se lleva en el lado del servidor.
- ii. **Bajo coste en actualizar los equipos con una nueva versión:** Los navegadores web visualizan (renderizan) las páginas web que son servidas por el servidor web dinámicamente. En ese sentido, es el servidor quien ejecuta la mayor parte del código de la aplicación, y suministra de forma centralizada las vistas las páginas a los navegadores conectados. En consecuencia, no hay que instalar nada en los clientes, ya que la actualización se realiza en el servidor, y automáticamente la ven todos los usuarios.
- iii. **Acceso a la última y mejor versión:** Se evita que pueda existir algún equipo que ejecute una versión diferente y desactualizada. Si existen ordenadores con distintas versiones del programa se puede originar problemas de consistencia en la información, o pérdida de funcionalidad. Todos los usuarios utilizan obligatoriamente la versión más actualizada del software.
- iv. **Información centralizada:** En una aplicación web, no sólo la lógica de negocio está centralizada en el servidor, sino también los datos que se ubican en una base de datos centralizada, como puede observarse en la *Ilustración 1*. La centralización tiene la ventaja de facilitar el acceso a la misma.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

- v. **Seguridad y copias de seguridad:** Con la disponibilidad de los datos centralizada es más fácil establecer y llevar el control de una política de copias de seguridad centralizada.
- vi. **Movilidad:** El software está ubicado en un servidor web en Internet y, por lo tanto, cualquier usuario con un dispositivo adecuado y una conexión a internet, podría acceder a la aplicación.
- vii. **Reducción de costes en los puestos cliente:** Debido a que las páginas se ofrecen desde el servidor web, quien suele ejecutar la mayoría de los procesos y la lógica de negocio, el equipo cliente queda relegado a mostrar los resultados y formularios, para lo cual no es necesario un hardware potente en la máquina cliente, lo que se traduce en reducción de costes y una mayor longevidad.



Ilustración 1 – Arquitectura web: cliente-servidor- base de datos

Ventajas de las aplicaciones de escritorio [Moreno, J.M. (2011)]:

- i. **Permite un mejor aprovechamiento del hardware y software del equipo:** Las aplicaciones de escritorio suelen estar desarrolladas pensando en un sistema operativo y familia hardware concretos. Estas aplicaciones tienen un mayor acceso al hardware del equipo por lo que podrán realizar acciones a día de hoy imposibles mediante un software web. Aunque ciertas aplicaciones web intentan resolver estos inconvenientes mediante “plugins” que se ejecutan en el lado del cliente, ciertamente están muy



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

limitadas en el acceso al hardware del equipo cliente. Por otro lado, las mayoría de las aplicaciones web están desarrolladas con lenguajes interpretados, los cuales tienen como inconveniente un peor rendimiento (ej: html, php, javascript, actionscript, java) son todos interpretados o pseudointerpretados.

- ii. **Polifacético:** El software “tradicional”, debido a que se ejecuta en su mayoría en el lado del cliente y tiene un mayor acceso al hardware puede abordar un mayor tipo de aplicación, desde aplicaciones industriales de control y monitoreo de procesos, hasta accesos a bases de datos o videojuegos. En ese sentido, las aplicaciones web, están bastante más limitadas, y por tanto existen ámbitos que a día de hoy no pueden resolver.

- iii. **Menores tiempos de desarrollo y menor coste:** Las aplicaciones de escritorio cliente/servidor llevan en el mercado mucho más tiempo que las aplicaciones web. Por esa razón, las herramientas de desarrollo están más evolucionadas y depuradas para el desarrollo de aplicaciones de escritorio en la creación interfaces de usuario o conectar con bases de datos, permitiendo al desarrollador ahorrar tiempos. Otro aspecto importante es la depuración de las aplicaciones, es decir, el monitoreo para la detección y corrección de errores, los cuales están muy bien resueltos en aplicaciones “tradicionales”. Por el contrario, el desarrollo de aplicaciones web, viene derivado del diseño de páginas web, es decir, de la época en la que las páginas web eran estáticas, y el lenguaje de descripción está pensado para maquetar un diseño gráfico. Para dotarle de dinamismo, el programador debe apoyarse en lenguajes de programación en el lado del servidor (ej. java, php, asp, etc) y otros en el lado del cliente (javascript, actionscript, vbscript, etc), lenguajes diferentes que complican el desarrollo (el programador debe trabajar con varios lenguajes distintos, a diferencia del software tradicional),



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

y las herramientas de desarrollo aún distan bastante del software de escritorio.

- iv. **Mejores tiempos de respuesta:** Como consecuencia del mejor aprovechamiento del hardware y software del equipo, se pueden conseguir unos tiempos de respuesta menores. Esto es algo que puede ser vital según el caso.
- v. **No dependencia de internet:** Habitualmente las aplicaciones de escritorio son cliente/servidor en red local, de modo que no es necesario el uso ni la dependencia de internet para trabajar.
- vi. **Si se desea y el software lo permite, puede trabajar también por internet:** La ventaja de trabajar online, no es exclusiva del software web. Las aplicaciones cliente/servidor pueden trabajar con este modelo, con independencia de si el servidor está en tu oficina u otra ciudad, empleando internet como medio de comunicación y siguiendo de nuevo la estructura de la *Ilustración 1*. Por contra, la inmensa mayoría de las aplicaciones web se ofrecen como un servicio externalizado por lo que existe una total dependencia de internet.
- vii. **Siempre funcionará, sin dependencia del navegador web:** Las aplicaciones web, tienen una total dependencia del navegador web. Aunque suelen desarrollarse utilizando un estándar internacional (HTML del W3), no todos los navegadores lo implementan al 100%. Eso conlleva a que a la hora de desarrollar se deban fijar unos navegadores como objetivo y revisar su funcionamiento en estos. Sin embargo existe efectos colaterales. Los navegadores suelen actualizarse con bastante frecuencia por razones de seguridad, e incorporando nuevas funcionalidades. Por esa razón, no es extraño ver que tras una actualización del navegador, lo que antes funcionaba correctamente ahora deja de hacerlo o se muestra incorrectamente, lo cual puede requerir la constante



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

revisión del programa, lo cual se traduce en tiempos y costes.

En la actualidad se observa claramente un gran crecimiento de las aplicaciones de tipo web, mientras que las aplicaciones de escritorio están más estancadas. Las aplicaciones de escritorio son más antiguas, llevan décadas en desarrollo, de ahí que se las denomine como “tradicionales”. Mientras que las aplicaciones web son más “jóvenes” y llevan pocos años evolucionando. Ésta es una de las razones de que las aplicaciones web estén en auge ahora mismo.

¿Podemos decir entonces que las aplicaciones web son mejores que las aplicaciones de escritorio? La respuesta es “no”. Ya se han mencionado las ventajas y desventajas de cada una, y a través de ellas no podemos determinar qué solución es mejor. Lo importante es decidir qué tipo de aplicación conviene a nuestro proyecto atendiendo a sus necesidades. Sí existe una mejor solución para un determinado proyecto.

Centrándonos en la aplicación que se desea desarrollar en este trabajo se ha optado por una solución de tipo escritorio. El procesado de imágenes necesario para la aplicación no es factible con una solución web, sin embargo el mayor aprovechamiento del software y hardware del equipo por parte de una aplicación “tradicional” decanta bastante la balanza hacia este lado. La característica polifacética de las aplicaciones de escritorio permiten un mayor margen de maniobra para desarrollar las tareas que se requieren, y la independencia de navegador e internet también son una importante ventaja para este tipo de proyecto. Y, aún así, permite la conexión a internet para determinadas tareas. Por el contrario la instalación, versiones, actualizaciones y movilidad suponen un mayor problema en esta alternativa.

1.2. Sistemas operativos

Toda aplicación debe ejecutarse sobre un sistema operativo, ya que es este software el que gestiona los recursos de hardware y proporciona los servicios necesarios a los programas. Los tres sistemas operativos más utilizados son Linux, Mac y Windows.

Para el desarrollo de la aplicación Autodesign el sistema operativo elegido ha sido Windows, debido a que es el sistema más extendido y habitual. De esta



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

forma, en el caso de haber usuarios finales, se podría distribuir a una mayoría de ellos.

2. CONCEPTOS DE IMAGEN

Para el desarrollo de una aplicación de tratamiento de imagen, es evidente que el principal tipo de datos con el que se va a trabajar es la imagen. Por tanto, es necesario conocer los distintos tipos de imágenes digitales que se pueden manejar y sus formatos. Conociendo sus características podemos aprovechar las ventajas que cada formato nos ofrece.

2.1. Tipos de imagen digital

Existen dos principales tipos de imágenes digitales, con los que se trabaja de forma completamente diferente. Estos son las imágenes vectoriales y las imágenes de mapa de bits, las cuales se explican a continuación [Arribas, A.G. (2014)]:

Imágenes de mapa de bits: Están formadas por una serie de puntos (píxeles), cada uno de los cuales contiene información de color y luminosidad.

Las imágenes de mapa de bits presentan una mayor gama de colores y de tonos que las vectoriales, por lo que son el tipo de imágenes usado en fotografía y, se crean con las cámaras de fotos, los escáneres y con programas de edición de imagen y dibujo (Adobe Photoshop, Gimp, etc.) Una desventaja es que estos archivos ocupan mucha más memoria que las imágenes vectoriales.

Imágenes vectoriales: Están constituidas por objetos geométricos autónomos (líneas, curvas, polígonos,...), definidos por ciertas funciones matemáticas (vectores) que determinan sus características (forma, color, posición,...). Se crean con programas de diseño o dibujo vectorial (Adobe Illustrator, Corel Draw, Inkscape...) y suelen usarse en dibujos, rótulos, logotipos...



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Su principal ventaja es que una imagen puede ampliarse sin sufrir el efecto de “pixelado” que tienen las imágenes de mapa de bits al aumentarse.

Algunos formatos de imagen vectorial son: AI (Adobe Illustrator), CDR (Corel Draw), DXF. (Autodesk), EMF, EPS, ODG (Open Office Draw), SVG (Inkscape), SWF (Adobe flash), WMF (Microsoft).

En la imagen (vectorial) del ratón, situada a la izquierda de la *Ilustración 2*, puede apreciarse que al ampliar una zona no hay pérdida de detalle, mientras que en la fotografía del busto Nefertiti (mapa de bits) de la misma ilustración, al ampliar mucho una zona, se observan los píxeles y la imagen se degrada.



Ilustración 2 – Imagen vectorial / imagen en mapa de bits

Las imágenes base de las que parte el proyecto están compuestas por mapas de bits, ya que provienen de fotografías hechas con cámaras digitales. Esto fuerza que el tipo de imágenes elegido para trabajar en la aplicación sea este. Aún así, para el procesado de fotografías, como es el caso, es más adecuado, y sencillo, trabajar con mapas de bits que con imágenes vectoriales. Habrá que prestar una especial atención al peso o tamaño de estos archivos de imagen durante el desarrollo de la aplicación.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

2.2. Formatos de imagen

Centrándonos en las imágenes basadas en mapas de bits existe un amplio rango de posibilidades en cuanto a formatos. En esta sección, se analizan los principales atendiendo a las características más importantes que los diferencian; tales como calidad de imagen, compresión, profundidad de color, tamaño...

Antes de estudiar los diferentes formatos, hay que tener claros los siguientes conceptos [Hernández, J.R. (2010)], [Mateos, M. (2011)], [Moreno, J.M. (2010)]:

Contenido de la imagen: Puede ser foto, dibujo, gráfico, logotipo... Muy importante a la hora de decidir que tipo y formato de imagen utilizar.

Calidad: La calidad que se desea obtener en función de su destino: publicación en la web impresión en impresora doméstica, impresión profesional,...

Tamaño: La cantidad de bytes que ocupará en memoria el archivo resultante. Los formatos pueden ser más o menos eficientes, pero el peso o tamaño de la imagen siempre estará reñido con la calidad de la imagen.

Profundidad de color: Es la cantidad de bits de información necesarios para representar el color de un píxel. A mayor profundidad de color, más detalle y calidad de color, pues se pueden representar un número mayor de posibles colores.

Compresión: Consiste en la reducción del peso o tamaño de los archivos de imagen mediante la aplicación de un algoritmo que recodifica la imagen a un código de menor tamaño que el original. Se pueden realizar compresiones de dos tipos:

- i. **Con pérdida:** Estos formatos no guardan toda la información que tiene la imagen original. Se realiza una compresión descartando cierta información considerada como "*irrelevante*" por tener demasiado detalle para el ojo humano o, simplemente, por simplificación. De ese modo se reduce el tamaño de los archivos, con una calidad razonable para ciertos tipos de imágenes.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Es perfecto para distribuir y almacenar fotografías, webs o imágenes con muchas texturas. Sin embargo no es un formato ideal para imágenes con muchas zonas con color plano, o muchos bordes (como por ejemplo, aquellas que tienen mucho texto, particularmente con tipografías delgadas). En la *Ilustración 3*, se puede observar la calidad de una imagen con distintas tasas de compresión.

- ii. **Sin pérdida:** Al contrario que los anteriores, no descartan información, sino que la almacenan y/o comprimen, de forma que siempre será posible reconstruir la imagen original. Se aplica para ello una compresión que no elimina información de la imagen original y, en consecuencia, la reducción de tamaño será menor que en el caso anterior.

Este tipo de formatos no debe utilizarse para fotografías o imágenes con muchas texturas y colores, sino para imágenes con colores sólidos: dibujos, gráficas, capturas de ventanas del sistema operativo, imágenes lineales,...



Ilustración 3 – Compresión con pérdida JPEG



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Como ya se ha mencionado, el tamaño está ligado con la calidad, y por lo tanto una mayor compresión conlleva también una mayor pérdida de calidad. Se pueden utilizar distintos tamaños de compresión, expresados en porcentaje de calidad con respecto al tamaño original.

Principales formatos de imagen basados en mapa de bits [Hernández, J.R. (2010)], [Mateos, M. (2011)]:

BMP (Bitmap = Mapa de bits): Formato introducido por Microsoft y usado originariamente por el sistema operativo Windows para guardar sus imágenes. Su mayor ventaja es que no sufre pérdidas de calidad porque apenas tiene compresión y por tanto resulta adecuado para guardar imágenes que se quieran editar a posteriori. La desventaja es que al guardar toda la información de cada pixel, es un formato bastante pesado. Es un formato algo anticuado.

GIF (Graphics Interchange Format = Formato de Intercambio Gráfico): Formato bastante antiguo desarrollado por CompuServe con el fin de conseguir archivos de tamaño muy pequeños. Admite solo 256 colores (profundidad de color de 8 bits) por lo que no es adecuado para imágenes fotográficas pero si es muy apropiado para imágenes en webs y logotipos, dibujos, etc. Permite crear animaciones y transparencias.

JPG-JPEG (Joint Photographic Experts Group = Grupo de Expertos Fotográficos Unidos): Es uno de los formatos más conocido y usado para fotografías digitales ya que admite millones de colores. Lo admiten la mayor parte de las cámaras fotográficas y escáneres y es muy utilizado en páginas web, envío de fotografías por correo electrónico, presentaciones multimedia y elaboración de vídeos de fotografías.

JPEG admite distintos niveles de compresión con pérdida, de forma que podemos optar por distintos niveles de compresión según nuestras necesidades de tamaño y calidad.

Cada vez que se abre y manipula una foto JPEG en un ordenador, la imagen al comprimirse y descomprimirse se degrada, por lo que conviene no guardarlas en JPEG si se van a modificar. En este caso usar TIFF o BMP para editarlas y convertirlas a JPEG al final. Si no queda más remedio que editar en JPEG, manipularlas con cuidado y no excesivamente.

TIF-TIFF (Tagged Image File Format = Formato de Archivo de Imagen Etiquetada): Es un formato similar a BMP, pero que añade compresión

CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

con pérdidas y algunas nuevas características. Muy utilizado para el escaneado, la edición e impresión de imágenes fotográficas. Es compatible con casi todos los sistemas operativos y editores de imágenes. Admite millones de colores (gran profundidad de color), capas, canales alfa... y también lo incluyen algunas cámaras y la mayoría de los escáneres. El excesivo tamaño de las imágenes en este formato ha producido que no esté tan extendido como otros.

PNG (Portable Network Graphic = Gráfico portable para la red): Formato creado con el fin de sustituir a GIF. Utiliza sistemas de compresión gratuitos, y admite muchos más colores que GIF. Nace como sustituto del GIF. Permite la compresión de archivos de 24 bits sin pérdida y la de archivos de mayor calidad con pérdida de hasta 24 bits. Ofrece una buena calidad de imagen aunque quede lejos de formatos más profesionales como puede ser TIF. No obstante se ha de tener en cuenta que las versiones más antiguas de los navegadores no pueden reconocer este tipo de archivos. Igual que el GIF, puede presentar transparencia, sin embargo no permite la inclusión de animaciones. Al admitir más colores es posible crear imágenes transparentes con mayor detalle.



Ilustración 4 – Formatos básicos de imagen digital

RAW: Formato “en bruto” o “crudo”. Esto quiere decir que contiene todos los píxeles de la imagen captada, tal y como ha sido captada por el CCD (sensor) de las cámaras digitales de fotos. Es el formato que ofrece la mayor calidad fotográfica y suele ser admitido por cámaras de gama media y alta (réflex y compactas) indicadas para fotógrafos aficionados avanzados y profesionales.

Las cámaras que guardan las fotos en otros formatos (TIFF y JPEG) procesan la imagen captada para dar una interpretación de ella (balance



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

de blanco, niveles de luminosidad, contraste...). En el formato RAW, los píxeles no se procesan y se mantienen en bruto para ser procesados posteriormente por un software específico conocido como “revelador RAW”. Un archivo de este tipo, no sufre ninguna compresión, por lo que mantiene el máximo detalle de la imagen a costa de ocupar mucho espacio. Evidentemente, no utiliza ningún tipo de compresión, pues mantiene la imagen con la calidad original.

PSD: Es el formato de Adobe Photoshop. Los principales factores a su favor son la compresión sin pérdidas y la posibilidad de guardar imágenes manteniendo gran cantidad de información asociada, como las capas de imagen, las máscaras, etc. Lo utilizan mucho los diseñadores y fotógrafos y actualmente permite que se abra en otros programas de retoque. En algún momento ha llegado a tener incluso más relevancia que el formato TIFF, ya que hasta hace unos años este último no podía almacenar capas.

DNG (Digital Negative = Negativo Digital): Surge a raíz de intentar estandarizar los diferentes formatos RAW. Ofrece las numerosas ventajas de RAW y además permite generar modificaciones en el mismo formato DNG y deshacerlas si no estamos de acuerdo, cosa que no ocurre con el RAW. Una vez trabajadas las imágenes este formato permite almacenar en su interior vistas previas en JPEG, con lo que aumenta la velocidad en la previsualización del archivo.



Ilustración 5 – Formatos complejos de imagen digital



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

En la siguiente tabla se recogen de modo comparativo algunas de las características más importantes de los formatos estudiados [Valls Arnau, M. (2011)] que ayudarán con la decisión del formato.

Formato de imagen	Calidad en términos absolutos	Nivel de compresión	Compatibilidad para el intercambio de sistemas	Número de colores que puede usar
RAW	Ideal	Sin compresión	Muy limitada	48 bits
DNG	Ideal	Sin compresión	Limitada	48 bits
JPEG	Buena	Muy buena	Muy alta	24 bits
PNG	Buena	Buena	Alta	24 bits
TIFF	Muy buena	Regular	Muy alta	48 bits
PSD	Muy buena	Regular	Alta	48 bits
GIF	Baja	Buena	Alta	8 bits

Tabla 1– Comparativa de formatos de imagen

Dejando a un lado los formatos de imagen más complejos, la elección de un formato de imagen es una de las decisiones iniciales que puede cambiar el desarrollo del proyecto. Las citadas ventajas del formato TIFF tales como la calidad de imagen, la profundidad de color y la posibilidad de trabajar con transparencia son muy llamativas. Si a esto sumamos que nos permite trabajar con capas dentro de la propia imagen parece incuestionable que es la elección perfecta para este tipo de proyecto. Sin embargo, esto es solo teoría. En la práctica, surgieron problemas para tratar con este tipo de archivos. Las librerías del lenguaje de programación Python (se estudia detenidamente más adelante en este documento) no ofrecen el mismo rango de posibilidades que con formatos de imagen más comunes. Las limitaciones que produce el hardware también suponían un grave problema, pues el rendimiento que ofrecía mi computadora con la edición de estos archivos fue decepcionante. Si a esto sumamos el excesivo tamaño que ocupan estas imágenes, parecen motivos suficientes para buscar otra alternativa.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Prestando atención a la característica de transparencia en imágenes, ya que es un requisito imprescindible para la fusión de imágenes en una composición, el formato PNG aparece como la mejor alternativa. Ofrece una buena profundidad de color, con lo que nos permite trabajar imágenes con transparencia mejor que con otros formatos. La calidad de imagen no es la misma que la que ofrece un formato más profesional dedicado a la edición de fotografía, como pueden ser RAW o TIFF, pero es suficientemente buena para las necesidades de la aplicación. Utilizando una mínima compresión (con leves pérdidas de calidad) se ha obtenido imágenes de calidad adecuada en archivos de imagen de tamaño bastante más reducido que los citados anteriormente. No debemos olvidar que la aplicación está enfocada para su distribución a usuarios que utilizarán los formatos más habituales y extendidos, y PNG será el utilizado pues se requiere transparencia en las imágenes.

El formato PNG se ha utilizado para la composición de fotografías por planos combinados utilizando el canal alfa de las imágenes (explicado en el siguiente punto del documento); tanto con las propias de la aplicación como por las introducidas por los usuarios.

En la aplicación se hace uso de algunas imágenes que apenas requieren edición, pues no van a ser utilizadas para las composiciones fotográficas. Para éstas se ha elegido el formato JPEG, pues su compresión con pérdida es perfecta para estas imágenes que no requieren alta calidad, reduciendo así el peso de algunos archivos y, por tanto, el tamaño de la aplicación final.

Ejemplo de imagen utilizada en la aplicación: Plano de escenario;
resolución 1922 x 1440

i. Sin transparencia:

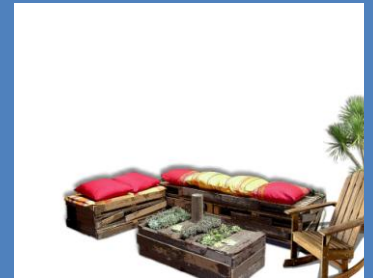


Ilustración 6 – Plano sin transparencia

Con transparencia:



Ilustración 7 – Plano con transparencia

Tamaños de archivo según el formato:

i. Sin transparencia:

BMP (color 32 bits):
10.812 KB

JPEG (compresión 50%):
337KB

ii. Con transparencia:

GIF (256 colores): 610 KB
TIFF (sin compresión, un plano): 13.201 KB

RAW (un plano): 10.812 KB

PSD (un plano): 5.062 KB
PNG (sin compresión):



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

2.3. Canales de color. El canal Alfa

Cuando las imágenes digitales pasan por algún tipo de procesamiento o modificación suelen ser divididas en distintos canales, que contienen distintos componentes de la imagen original. Cada canal es una imagen en escala de grises con, exactamente, el mismo tamaño que la imagen original y, en consecuencia, contiene el mismo número de píxeles. A cada uno de estos píxeles de la imagen en escala de grises se le puede considerar como un contenedor que se puede rellenar con un valor que indica la intensidad de la componente guardada en ese canal. Los modelos de descomposición más utilizados actualmente son RGB, RGBA, CMYK, YUV y HLS [Davidson, S.R. (2009)]:

RGB: Estas siglas provienen del inglés red, green, y blue; esto es, rojo, verde, y azul, respectivamente. Este trío de colores intenta modelar el sistema visual humano. La mayoría de los monitores y televisores se basan en este modelo, debido a que están basados en la emisión de luz. Este modelo también se le atribuye la propiedad de sistema aditivo, ya que se añaden las intensidades para formar un color.

RGBA: Se trata del mismo modelo RGB, pero con otra propiedad: canal alfa. Es posible ver la incorporación de este canal a una imagen RGB en la *Ilustración 8*. Este canal se usa como un índice de la transparencia en un píxel. Esto nos sirve a la hora de mezclar varios colores designados para un solo píxel. Este modelo es más reciente y se suele usar para crear efectos y técnicas visuales como el suavizado de imagen (o "anti-aliasing"), niebla, llamas, y objetos semi-transparentes: cristal, agua, vidrieras, etcétera.

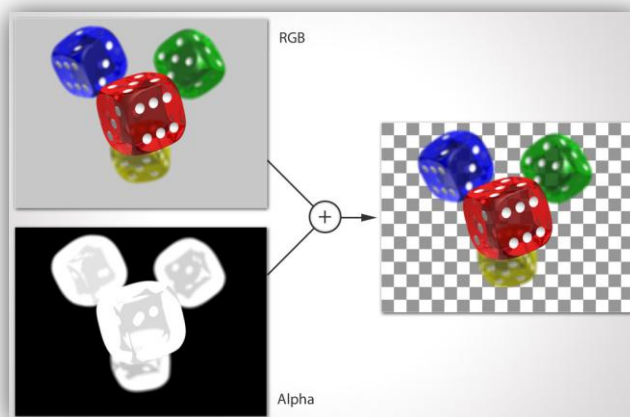


Ilustración 8 – Canales RGB + canal Alfa = RGBA

CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

CMYK: Las siglas representan, en inglés, cyan, magenta, yellow y black; esto es, cian, magenta, amarillo, y negro, respectivamente. Los tres primeros colores son complementarios a los de RGB. En la *Ilustración 9*, se puede apreciar la diferencia entre ambos modelos. Este modelo se usa más en la imprenta para crear imágenes a partir de tintas de colores. Para obtener un color más vivo, se le añade la tinta negra también. Este modelo tiene la propiedad de sistema sustractivo, ya que se restan las intensidades a la luz. El pigmento se obtiene porque el material absorbe la energía de la luz y por tanto una parte de su espectro. La energía que transmite se percibe como un color determinado. Por ejemplo, nosotros percibimos el color verde, porque tal pigmento absorbe todos los "colores" del espectro de la luz emitida y transmite/refleja el color verde.



Ilustración 9 – Canales RGB / Canales CMYK



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

HLS: En inglés, las siglas son hue, lightness o luminance, y saturation; esto viene a ser, matiz, brillo, y saturación, respectivamente. Este modelo se basa en lo empírico de nuestra percepción visual. En lugar de usar un modelo tricolor para formar otros colores y tonos, el modelo HLS se basa en tres propiedades que sirven para definir los colores que percibimos. El matiz es el color que solemos denominar: azul, violeta, rojo, dorado, etcétera. El brillo describe la vividez y brillo de un tono de color. La saturación diferencia la percepción de un tono "puro" a otro tono que ha sido mezclado con blanco para formar un tono pastel - suave. Este modelo es usado por artistas, principalmente. Su comparación con respecto a los canales RGB se puede apreciar en la ilustración siguiente.



Ilustración 10 – Canales RGB / Canales HLS



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

YUV: También puede ser escrito como YCbCr o incluso YPbPr. Este modelo se basa en el modelo RGB, pero restringiendo y descomponiendo algunos valores del RGB, como puede observarse en

Ilustración 6. YUV se usa en señales de televisión y en equipos de vídeo y grabación como S-Vídeo, y MPEG-2 y por tanto DVD. Originalmente, la principal razón de usar este modelo fue para mantener compatibilidad con los televisores analógicos en blanco y negro, cuando se introdujeron los televisores en color. El componente Y es el brillo total. U y V son componentes cromáticos (colores) basados en los componentes rojo y azul del modelo RGB. El proceso comienza con una imagen en blanco y negro (Y). Luego, se obtiene el componente U, mediante una resta entre Y y el componente azul del RGB original, y V mediante una resta entre Y y el componente rojo.

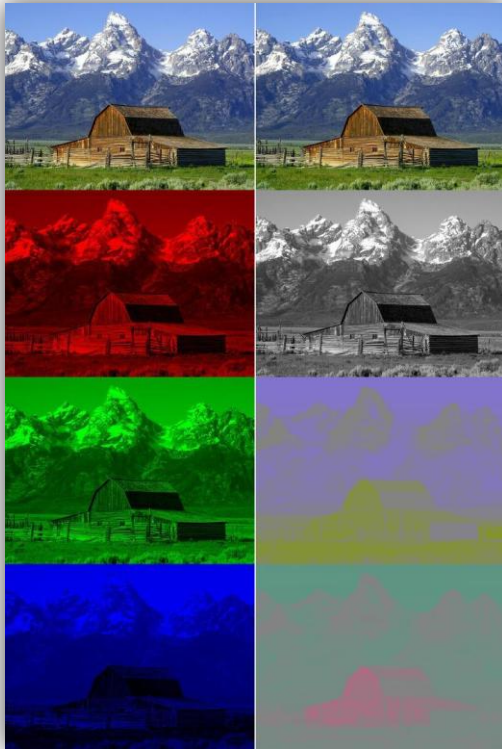


Ilustración 11 – Canales RGB / Canales YUV

El canal alfa: El canal alfa de una imagen 2D es el almacenamiento de un valor de transparencia u opacidad para cada pixel. De esta forma, este canal permite definir el grado de transparencia que tendrá cada pixel de una imagen en mapa de bits: Esta transparencia puede ser de opacidad total, transparencia total, o cualquier valor intermedio entre ambas. Puede estar expresado en valores entre 0 (transparente) y 1 (opaco), entre 0 y 255 (opaco)...

De esta forma, una máscara de transparencia u opacidad no es más que una capa que se aplica



Ilustración 12 – Máscara de opacidad



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

a una imagen sin transparencia definiendo los distintos valores opacidad para la imagen. En la *Ilustración 12* se puede ver un claro ejemplo del uso de una máscara de opacidad.

Este concepto resulta clave para el desarrollo del proyecto, pues en él se basa todo el proceso de apilado de capas para lograr el resultado de una composición final. Además, los valores intermedios de transparencia permiten crear degradados de opacidad, permitiendo una fusión de imágenes de forma suavizada evitando el aliasing (técnica anti-aliasing).

3. TECNOLOGÍAS

3.1. Lenguajes de programación: Python

Un lenguaje de programación es un conjunto de órdenes que un equipo puede ejecutar. De esta forma los seres humanos podemos comunicarnos con el hardware y software de un equipo e indicar las instrucciones y tareas que queremos que desarrollen.

Gracias a los lenguajes de programación podemos especificar, de forma precisa, cuáles son los datos que debe manipular, de qué modo deben ser almacenados, transferidos o procesados dichos datos y qué instrucciones debe poner en marcha la computadora ante ciertas circunstancias.

Existen distintos tipos de lenguajes de programación atendiendo al nivel de abstracción de estos, desde los de más bajo nivel hasta los de más alto nivel de abstracción. Es decir, desde los que más se asemejan al lenguaje propio de la computadora (lenguaje de máquina) hasta los que más se parecen a los lenguajes humanos.

También existe una distinción de lenguajes en base al procesamiento de sus comandos [Kioskea. (2014)]:

Lenguajes imperativos: Un lenguaje imperativo programa mediante una serie de comandos, agrupados en bloques y compuestos de órdenes condicionales que permiten al programa retornar a un bloque de comandos si se cumple la condición. Estos fueron los primeros lenguajes de programación en uso y aún hoy muchos lenguajes modernos usan este principio.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

No obstante, los lenguajes imperativos estructurados carecen de flexibilidad debido a la secuencialidad de las instrucciones.

Lenguajes funcionales: Un lenguaje de programación funcional (a menudo llamado lenguaje procedimental) es un lenguaje que crea programas mediante funciones, devuelve un nuevo estado de resultado y recibe como entrada el resultado de otras funciones. Cuando una función se invoca a sí misma, hablamos de recursividad.

Pero existen, además, dos principales categorías de lenguajes de programación [Kioskea. (2014)]:

Lenguajes interpretados: Un lenguaje de programación es, por definición, diferente al lenguaje máquina. Por lo tanto, debe traducirse para que el procesador pueda comprenderlo. Un programa escrito en un lenguaje interpretado requiere de un programa auxiliar (el intérprete), que traduce los comandos de los programas según sea necesario.

Lenguajes compilados: Un programa escrito en un lenguaje "compilado" se traduce a través de un programa anexo llamado compilador que, a su vez, crea un nuevo archivo independiente que no necesita ningún otro programa para ejecutarse a sí mismo. Este archivo se llama ejecutable.

Un programa escrito en un lenguaje compilado posee la ventaja de no necesitar un programa anexo para ser ejecutado una vez que ha sido compilado. Además, como sólo es necesaria una traducción, la ejecución se vuelve más rápida. Sin embargo, no es tan flexible como un programa escrito en lenguaje interpretado, ya que cada modificación del archivo fuente (el archivo comprensible para los seres humanos: el archivo a compilar) requiere de la compilación del programa para aplicar los cambios. Por otra parte, un programa compilado tiene la ventaja de garantizar la seguridad del código fuente.

Lenguajes intermediarios: Algunos lenguajes pertenecen a ambas categorías (LISP, Java, Python...) dado que el programa escrito en estos lenguajes puede, en ciertos casos, sufrir una fase de compilación intermedia, en un archivo escrito en un lenguaje ininteligible (por lo tanto diferente al archivo fuente) y no ejecutable (requeriría un intérprete).



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Con esto se plantea otra decisión importante, la elección de un lenguaje de programación. Vamos a analizar algunos puntos clave que se deben tener en cuenta para abordar este problema.

No todos los lenguajes son igualmente válidos para realizar cualquier tipo de proyecto, sino que por sus características son más adecuados para determinados fines. Algunos incluso son utilizados para el desarrollo de sistemas operativos, como puede ser C/C++.

En la actualidad el desarrollo web y las aplicaciones móviles están obteniendo una gran importancia. HTML es la base la programación, donde se apoyan otros lenguajes como CSS para aportar el estilo, PHP o JSP para la creación de sitios web dinámicos desde el lado del servidor y Javascript desde el lado del cliente. Python o Ruby son lenguajes bastante nuevos y, con sus respectivos *frameworks* para web Django y Ruby on Rails, también permiten la creación de sitios web.

Para el desarrollo de aplicaciones móviles orientadas a móviles y tables se puede optar por la programación en código nativo, es decir, en el código en el que fueron desarrollados los sistemas operativos móviles. De esta forma se pueden desarrollar aplicaciones para iOS en Objective C o Swift, para Android en Java o para Windows Phone en C#. Por otro lado también se puede optar por desarrollar aplicaciones web basandose en los lenguajes mencionados anteriormente, así como optar por las soluciones híbridas, que agrupan ventajas de las dos anteriores.

Si nuestra intención es dedicarnos a la producción de videojuegos o 3D deberíamos centrarnos en lenguajes C/C++, C# o Java.

Java, C, C++, Python y Ruby pueden ser buenas opciones para programas de escritorio. Los tres primeros llevan bastante tiempo utilizándose para este propósito, por lo que será siempre una solución acertada, aunque también son más complejos pues son de más bajo nivel. Python y Ruby son los lenguajes de programación de “scripting” más utilizados y además poseen una mejor curva de aprendizaje.

También es importante analizar la popularidad y la oferta de trabajo antes de lanzarnos a aprender un nuevo lenguaje. A continuación se muestran una tabla (*Tabla 2*) y un gráfico (*Gráfico 1*) sobre el índice de popularidad de los distintos lenguajes, cuyos datos han sido extraídos del PYPL o Índice de Popularidad de los Lenguajes de Programación [J]. Éste se elabora analizando el número de búsquedas en Google de tutoriales sobre un determinado lenguaje.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

En la tabla se recogen los índices de popularidad de los principales lenguajes de programación en el mes de Junio de 2015, y con respecto a Junio de 2014

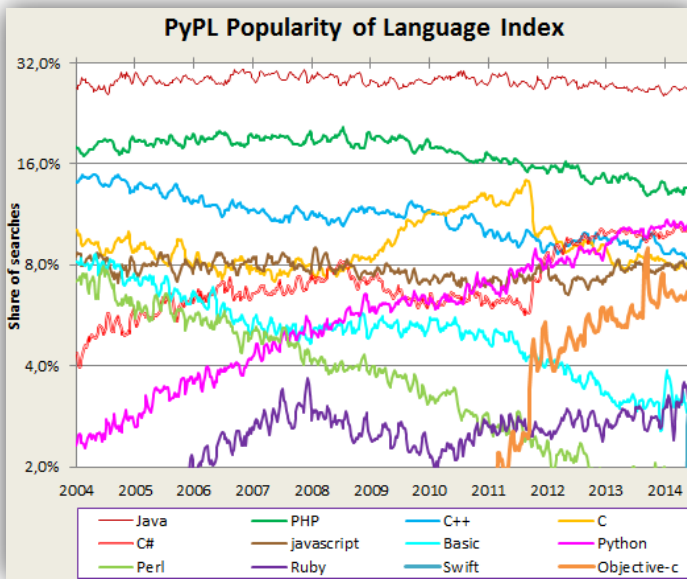


Gráfico 1 – Índices de popularidad de los lenguajes de programación 2004-2014

Rank	Change	Language	Share	Trend
1		Java	24.4 %	-0.9 %
2		PHP	11.5 %	-1.3 %
3		Python	10.9 %	+1.5 %
4		C#	9.1 %	-0.6 %
5		C++	7.8 %	-0.1 %
6		C	7.2 %	-0.2 %
7		Javascript	7.1 %	-0.5 %
8		Objective-C	5.5 %	-0.4 %
9		Matlab	3.0 %	+0.4 %
10	↑↑	R	2.8 %	+0.7 %
11	↑↑↑↑↑	Swift	2.7 %	+2.3 %
12	↓	Ruby	2.5 %	-0.0 %
13	↓↓↓	Visual Basic	2.1 %	-0.5 %
14	↓	VBA	1.5 %	-0.1 %
15	↓	Perl	1.2 %	-0.3 %
16	↓	lua	0.5 %	-0.0 %

Tabla 2 – Índices de popularidad de los lenguajes de programación

El siguiente gráfico muestra el porcentaje de ofertas de empleo registradas para los diferentes lenguajes de programación:

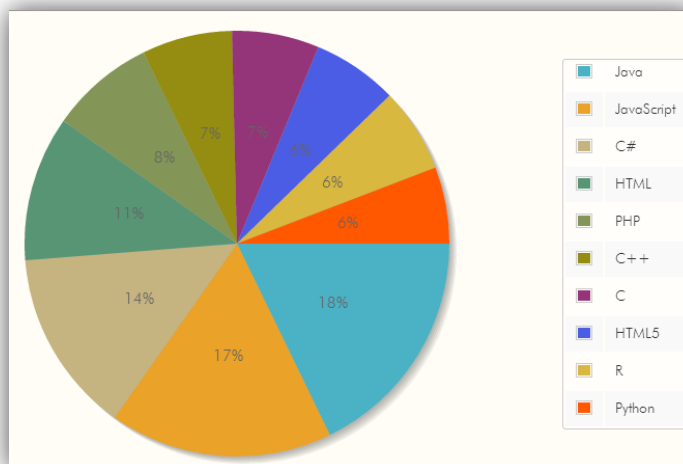


Gráfico 2 – Oferta de empleo según los lenguajes de programación

Porcentaje de ofertas de empleo según el lenguaje de programación en el mes de Mayo de 2015 en Estados Unidos y Europa según la herramienta web TrendySkills disponible en <http://trendyskills.com/>



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Partiendo de la idea de desarrollar una aplicación de escritorio, Java parte con ventaja pues el lenguaje del cual más conocimiento tengo. Python y Ruby captaron mi interés por ser lenguajes más modernos y en crecimiento, por lo que les veo más futuro. Sus posibilidades, sencilla sintaxis y buena curva de aprendizaje (gracias a ser de alto nivel) decantan la balanza por aprender uno de estos lenguajes. Finalmente, la versatilidad y polivalencia de Python, sus librerías (siguiente apartado del documento) y robustez hicieron que me decantara por este lenguaje. Aún así, creo que Java también habría sido una opción perfectamente válida. Ruby podría haber funcionado, pero parece ser que donde despliega su auténtico potencial es en el uso del *framework* web Ruby on Rails.

Python: Es un lenguaje de "scripting" exitoso. Éste fue inicialmente desarrollado por Guido van Rossum [González Duque, R. (2007)]. Python fue por primera vez liberado en 1991. Es un lenguaje interpretado, de alto nivel, de propósito general, multiplataforma y de fácil aprendizaje. Una de sus características más visibles es que no utiliza comas ni paréntesis, utilizando la sangría en su lugar. Esto favorece la



Ilustración 8 – Logo de Python

estructuración y legibilidad del código en gran medida. En la actualidad Python es mantenido por un numeroso grupo de voluntarios en todo el mundo. Es polivalente, pues permite desarrollar aplicaciones web, aplicaciones de escritorio, interfaces de usuario, análisis de datos, cálculo de estadísticas,... utilizando los *frameworks* adecuados para Python.

3.2. Librerías de Python

La versatilidad que ofrece este lenguaje de programación viene dada en gran parte por la gran cantidad de funcionalidades base incorporadas en el lenguaje, permitiendo acometer casi cualquier tipo de proyecto. Estas librerías estándar contienen todos los módulos básicos que podamos necesitar, sacándonos de apuros en numerosas ocasiones. Si a esto añadimos las librerías de terceros que podemos incorporar, el abanico de posibilidades se multiplica aún más si es posible.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Durante el desarrollo se han utilizado algunas de estas librerías externas, todas ellas bastante habituales en muchos proyectos Python. Para el uso de estas librerías externas es necesario un proceso de descarga e instalación para poder comenzar a utilizarlas. Este proceso viene detallado en el anexo disponible al final del documento. Tres de estas librerías han sido fundamentales para el correcto funcionamiento de la aplicación, por lo que vamos a explicar brevemente en qué consiste cada una de ellas y cuáles son las posibilidades que nos ofrecen:

MySQLdb: Es un módulo externo que nos permite relacionarnos con bases de datos SQL a través de Python. Su uso es muy sencillo, pues incorpora funciones básicas para la conexión y desconexión con la base de datos, y las funciones de consultas de inserción, selección, modificación y borrado habituales de un lenguaje de bases de datos. Para hacer uso de ella únicamente debemos importar dicho módulo como se indica a continuación:

```
import MySQLdb
```

Python Image Library (PIL): Como su propio nombre indica, esta librería se utiliza para la manipulación de imágenes con Python. Soporta múltiples formatos de imagen y proporciona capacidades de procesamiento de imágenes y gráficos bastante potentes [Linblad, Thomas, Kinser J.M. (2013)]. Contiene una gran diversidad de módulos de operaciones básicas de imagen, fotografía, filtrado, retoque fotográfico, dibujo...

Esta biblioteca ha permitido realizar prácticamente la totalidad de tareas necesarias en la manipulación de imágenes para lograr el resultado final de la aplicación

Se puede incorporar de forma completa o seleccionando alguna de sus módulos, ya que es bastante extensa:

```
import PIL
```

```
from PIL import ImageOps
```

WxPython: Es un kit de herramientas multiplataforma para la creación de Aplicaciones de escritorio GUI [ZetCode. (2011)]. El principal autor de wxPython es Robin Dunn. Con wxPython los desarrolladores pueden crear aplicaciones sobre Windows, Mac y sobre varios sistemas Unix.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

wxPython es una envoltura alrededor de wxWidgets, la cual es una biblioteca de herramientas C++ multiplataforma madura. Cónsta de cinco módulos básicos: *Controls*, *Core*, *Interfaz de Dispositivo Grafico (GDI)*, *Misc* y *Windows* [ZetCode, (2011)]. Para importar esta librería se puede hacer de la siguiente forma:

```
import wx
```

También es posible importar un módulo específico:

```
import wx.media
```

Esta biblioteca es la que nos va a permitir crear una interfaz gráfica con la que el usuario pueda llevar el control de lo que sucede dentro de la aplicación. Es muy completa, pues contiene prácticamente todos los elementos básicos que se puedan necesitar para crear la interfaz de una aplicación de escritorio.

Esta última ha sido elegida entre las principales bibliotecas dedicadas a la interfaz gráfica de usuario en Python: tKinter, WxPython, PyQt y PyGTK. tKinter es la única que viene preinstalada con Python, es la más básica y fácil de aprender. PyQt y PyGTK son algo más complejas y, por lo tanto, más difíciles de aprender. Son bastante completas y flexibles, pero disponen de una documentación bastante reducida. Por último, wxPython es bastante completo, flexible, rápido, extendido, semejante al propio Python y con una documentación más completa. Estos factores me hicieron decantarme por esta alternativa, a pesar de ser relativamente más difícil de aprender debido a su complejidad. En mi opinión, una biblioteca más completa asegura tener un mayor número de soluciones a los problemas que van surgiendo durante el desarrollo.

3.3. Herramientas de desarrollo: Eclipse

Las herramientas de desarrollo son aquellas que nos van a facilitar los medios necesarios para producir nuestra aplicación. A continuación se definen una serie de conceptos que conviene tener claros a la hora de hablar de este tipo de herramientas [Luis, A. (2010)]:

IDE: Son las siglas de Entorno de Desarrollo Integrado. Es una aplicación (entorno de programación) para desarrollar software que está compuesto normalmente por un editor de código, un compilador o un



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

intérprete, herramientas auxiliares de desarrollo (creación de documentación, tests, empaquetado de software), un depurador y algunas veces un diseñador de interfaces gráficas incorporado.

GUI: Son las siglas de Interfaz Gráfica de Usuario. Son herramientas para crear interfaces gráficas, esto es, componentes gráficos con los cuales el usuario interactúa con la aplicación. Hay algunos IDE's que contienen GUI's integrados.

Editor: Es una aplicación para escribir código de programación en el lenguaje deseado, en nuestro caso, Python. Contiene todas las herramientas de edición necesarias, y algunas veces características propias de IDE's. Hay veces que cuesta distinguir entre un IDE y un editor por la cantidad de opciones que tiene este último.

RAD: Son las siglas de desarrollo rápido de aplicaciones. Dependiendo del autor sirve para designar a las aplicaciones de desarrollo de interfaces gráficas o a los IDE's con GUI's integrados, o a las dos cosas. Es un término dado a plataformas de desarrollo como PowerBuilder, Visual Studio o Delphi.

A la hora de seleccionar una herramienta de desarrollo debemos tener claro, como en otras ocasiones, que no existe la herramienta perfecta. Algunas herramientas pueden ser mejores que otras en algunos aspectos, pero a la hora de decidir debemos fijarnos en nuestro objetivo, así como en el nuestro gusto personal. Es conveniente probar algunas de ellas para ver si se adaptan a lo que realmente necesitamos y nos sentimos cómodos trabajando con ellas. Algunas opciones de herramientas son:

IDE con GUI integrado: BOA Constructor, SharpDevelop

IDE: IDLE, NetBeans, Stani's Python Editor, PyScripter, Geany, Pyragua, Wingware Python IDE, PythonWin, Eclipse.

GUI: wxGlade, wxFormBuilder

Editor: Editra, Notepad++, Jedit, Gedit, BlueFish, SciTE

Shell interactivo: Dreampie

Inicialmente mi idea era utilizar un IDE con GUI integrado, o un IDE y utilizar por separado una GUI. Una GUI que permitiera arrastrar gráficamente los componentes a introducir en la interfaz podría simplificar bastante el desarrollo. Probé algunos de estos, como BOA Constructor y WxGlade (los cuales me permitían trabajar con wxPython), sin llegar a convencerme.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

Rápidamente comprendí que para llevar un control de los componentes gráficos de usuario y lograr el dinamismo que pretendía en la aplicación debía introducirme de lleno en el kit de herramientas wxPython. Descarté la opción de utilizar una GUI y decidí empezar desde abajo con wxPython. Para ello decidí utilizar un IDE o un editor. Probé IDE que viene integrado con la instalación de Python, IDLE, así como con el editor Notepad++. Ninguno de ellos me pareció suficientemente completo, pues no contaban con todas las características propias de otros IDEs o RADs más completos. Finalmente, decidí hacer uso del IDE Eclipse, un entorno de desarrollo con el que ya estaba familiarizado. PyDev es un plugin de Eclipse que permite trabajar con el lenguaje Python. De esta forma Eclipse + PyDev formaron una herramienta suficientemente completa para el desarrollo el proyecto, teniendo en cuenta que no iba a hacer uso de una GUI. El proceso de instalación de Eclipse + PyDev viene detallado en el anexo del final del documento.

Con esto, la construcción del proyecto se apoya sobre tres pilares fundamentales: Eclipse, PyDev y wxPython; sin olvidar la base de todo el proyecto: el lenguaje de programación Python.



Ilustración 9 – Eclipse + PyDev + wxPython

CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

3.4. Servidor: XAMPP

Un servidor no es más que una computadora que se encarga de ofrecer unos servicios a otras computadoras que forman parte de un mismo sistema de red. Esta situación puede apreciarse en la *Ilustración 10* mostrada a continuación.

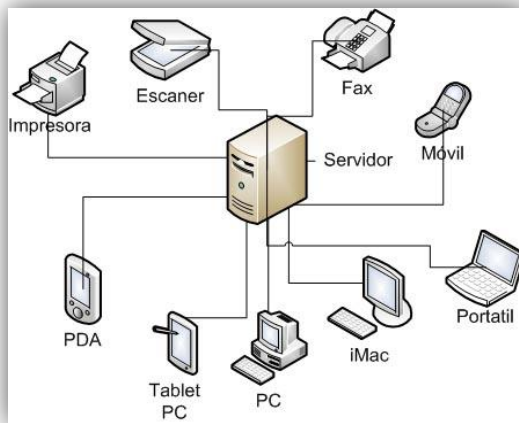


Ilustración 10 – Servidor en red

Básicamente, una computadora conectada a internet emplea una dirección (dirección web, dirección IP, dirección FTP, etc.) para poder comunicarse con el servidor al que le corresponde. La computadora envía (utilizando el protocolo adecuado) las distintas solicitudes al servidor, y el servidor responde (empleando el protocolo adecuado) las solicitudes. El servidor también puede solicitar datos de la

computadora, y la computadora le responde. Suelen estar dotados de una buena potencia de computación y buena conexión a Internet, ya que suelen satisfacer las peticiones de una importante cantidad de computadoras.

Existen diferentes servicios que pueden ofrecer los servidores dando lugar a los distintos tipos de servidores, los cuales se recogen en la *Tabla 5*, la cual se muestra a continuación [Sierra, M. (2006)]:

DENOMINACIÓN DEL SERVIDOR	DESCRIPCIÓN
Servidor de Correo	Es el servidor que almacena, envía, recibe y realiza todas las operaciones relacionadas con el e-mail de sus clientes.
Servidor Proxy	Es el servidor que actúa de intermediario de forma que el servidor que recibe una petición no conoce quién es el cliente que verdaderamente está detrás de esa petición.
Servidor Web	Almacena principalmente documentos HTML (son documentos a modo de archivos con un formato especial para la visualización de páginas web en los navegadores de los clientes), imágenes,



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

	videos, texto, presentaciones, y en general todo tipo de información. Además se encarga de enviar estas informaciones a los clientes.
Servidor de Base de Datos	Da servicios de almacenamiento y gestión de bases de datos a sus clientes. Una base de datos es un sistema que nos permite almacenar grandes cantidades de información.
Servidores Clúster	Son servidores especializados en el almacenamiento de la información teniendo grandes capacidades de almacenamiento y permitiendo evitar la pérdida de la información por problemas en otros servidores.
Servidores Dedicados	Hay servidores compartidos si hay varias personas o empresas usando un mismo servidor, o dedicados que son exclusivos para una sola persona o empresa.
Servidores de imágenes	Recientemente también se han popularizado servidores especializados en imágenes, permitiendo alojar gran cantidad de imágenes sin consumir recursos de nuestro servidor web en almacenamiento o para almacenar fotografías personales, profesionales, etc.

Tabla 3 – Tipos de servidores

Los servidores necesarios para obtener las funcionalidades del proyecto son el servidor de base de datos y el servidor web. La aplicación Autodesign funciona haciendo uso de tablas de información alojadas en una base de datos y, para poder acceder a dicha información, necesitamos un servicio de bases de datos. Además, para poder administrar la base de datos de una forma más cómoda será necesario utilizar un servicio web que nos ofrezca una página web dedicada a la gestión de nuestra base de datos.

A simple vista parece que utilizar un servidor de imágenes ofrece un servicio necesario para la aplicación, dado que se trata de un programa que debe almacenar una importante cantidad de imágenes. Sin embargo, se ha optado por trabajar con archivos de imágenes almacenados localmente para mejorar los



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

tiempos de respuesta de la aplicación. Por lo tanto un servidor de imágenes no será necesario para este caso.

Una vez claros los servicios que necesitamos en nuestro servidor analizamos las diferentes alternativas para montar un servidor en la computadora. Las tecnologías que nos permiten montar un servidor web son dependientes del sistema operativo en el que se implantan. Como ya se ha mencionado antes, estamos trabajando sobre el sistema operativo Windows y las principales alternativas para éste son [Hernández, J.R. (2013)]:

AppServ: Es una herramienta OpenSource para Windows que facilita la instalación de Apache, MySQL y PHP en la cual estas aplicaciones se configuran en forma automática. Como extra incorpora phpMyAdmin para el manejo de MySQL. La limitante de este software es la posibilidad de realizar configuraciones de manera sencilla (Cargar/Activar/Desactivar módulos para Apache o PHP). Con Appserv el control de servicios lo tendremos que hacer desde Herramientas administrativas del panel de control, ya que no cuenta con interfaz alguna para controlarlos. También como ventaja a mayores cabe mencionar que su proceso utiliza una reducida cantidad de memoria RAM.

XAMPP: Es uno de los más conocidos. Incorpora un servidor Apache, un sistema gestor de bases de datos MySQL y lenguajes como PHP y Perl. Además, ofrece soporte para gestionar cuentas FTP, acceso a bases de datos mediante PHPMyAdmin, bases de datos SQLite y varias otras características. También incluye un servidor de correos Mercury para el envío de emails, un servidor Tomcat para servlets JSP, y un servidor FTP FileZilla. Desde el panel de control se pueden gestionar fácilmente algunos de estos servicios, indicando cuales queremos que estén activos en cada momento. Es multiplataforma, por lo que funciona en sistemas Windows, Linux, Mac e incluso hasta Solaris. Para instalar XAMPP solo es necesario descargarlo de su página oficial y descomprimirlo, sin necesidad de hacer ningún cambio en los registros de Windows (portable).

WAMP: Se trata de un conjunto Apache + MySQL + PHP para Windows (WAMP), en el que además se incluye PHPMyAdmin para gestionar las bases de datos y webgrind como *profiler* para analizar el rendimiento de aplicaciones PHP5 en nuestro servidor. Su interfaz es muy simple y



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

sencillo, y tiene opción para establecer el idioma en español. De los tres es el que mayor configuración nos permite realizar y de una manera muy simple.

A partir del análisis de las tres soluciones para montar un servidor de forma local en la computadora, se concluye que XAMPP es la opción a utilizar. Haber trabajado anteriormente con WAMPP en la asignatura LDST (Laboratorio de Desarrollo de Sistemas Telemáticos) y con XAMP durante la realización de las prácticas en empresa propias del Grado permiten descartar fácilmente la opción de AppServer, pues me parecen ambas soluciones aceptables. Personalmente, me parece más cómodo trabajar con XAMPP puesto que tiene, en mi opinión, una interfaz más amigable y un panel de control (Ilustración 16) para activar y configurar cada uno de los servicios. Aún así, utilizar WAMP habría sido una opción perfectamente válida, ya que los servicios que ofrece son muy similares.



Ilustración 11 – Panel de control de un servidor XAMPP

En la sección de Anexos ubicada al final del documento se ofrece una explicación detallada de cómo configurar el servidor XAMP para poder utilizar sus servicios web y de base de datos de forma remota, es decir, desde cualquier computadora externa a la red de la computadora que monta el servidor. De esta forma es posible acceder a los servicios desde cualquier dispositivo con acceso a Internet y, obviamente, con los permisos necesarios.



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

3.5. Bases de datos

Concepto y tipos

Una base de datos (BD) es una herramienta que nos permite recopilar y organizar un conjunto de datos o información. Proporciona a los usuarios el acceso a la información, permitiendo visualizar, añadir, modificar o eliminar los datos. Se convierte más útil a medida que la cantidad de datos almacenados crece. Permite otorgar distintos permisos a los usuarios, de esta forma se puede gestionar las tareas permitidas para cada usuario.

Una base de datos puede ser local, es decir que puede utilizarla sólo un usuario en un equipo, o puede ser distribuida, es decir que la información se almacena en equipos remotos y se puede acceder a ella a través de una red. La principal ventaja de utilizar bases de datos es que múltiples usuarios pueden acceder a ellas al mismo tiempo.

Los principales tipos de bases de datos atendiendo a la estructura que presentan los datos son los siguientes [overblog. (2011)]:

Bases de datos jerárquicas: En desuso. Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

Las bases de datos en red: También en desuso. Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Bases De Datos Transaccionales: Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades. Estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial. Por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

Bases De Datos Relacionales: Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados *tuplas*. Se conceptualiza pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las *tuplas*, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Bases De Datos Multidimensionales: Son bases de datos ideadas para desarrollar aplicaciones muy concretas. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

Bases De Datos Orientadas a Objetos: Este modelo, bastante reciente, y propio de los modelos informáticos enfocado a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos.

Sistema Gestor de Bases de Datos (SGBD): MySQL

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos.

Los sistemas gestores de bases de datos más utilizados son MySQL, Oracle, Microsoft SQL Server y PostgreSQL. Todos ellos son sistemas de gestión de bases de datos relacionales orientados a objetos (ORDBMS, Object-Relational Data Base Management System). Los sistemas OO (Orientados a Objetos) almacenan la información sobre las relaciones de sus componentes junto a los datos y están diseñados para trabajar junto a lenguajes de programación OO. Oracle y Microsoft SQL Server son los más completos, pero por esta razón solo se permite trabajar con ellos bajo la compra de una licencia. MySQL y PostgreSQL son algo más sencillos y de licencia libre, y por ello, idóneos para este tipo de proyecto.

MySQL será la apuesta para este trabajo. Haber trabajado con él en anteriores proyectos y el hecho de venir integrado en el servidor elegido, XAMPP son motivos suficientes para descartar la alternativa de PostgreSQL



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

MySQL: Es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Entre sus ventajas destacan:

- i. Velocidad al realizar las operaciones.
- ii. Bajo costo en requerimientos para la elaboración de bases de datos.
- iii. Facilidad de configuración e instalación.

MySQL nos va a permitir gestionar la información de la aplicación en la base de datos. Nos ofrece la posibilidad de obtener los datos de forma organizada cuando los necesitemos, así como añadir, modificar o eliminar los datos. Más adelante en el documento se detalla la estructuración de los datos de la aplicación, llevada a cabo por este gestor.

Lenguaje de bases de datos: SQL

Un lenguaje de base de datos es un tipo de lenguaje de programación que nos permite relacionarnos con la base de datos. Estos lenguajes contienen un lenguaje de definición de datos, para especificar el esquema de la base de datos, y un lenguaje de manipulación de datos, para expresar las consultas a la base de datos y las modificaciones.

La manipulación de datos consiste en:

- i. La recuperación de información almacenada en la base de datos.
- ii. La inserción de información nueva en la base de datos.
- iii. El borrado de información de la base de datos.
- iv. La modificación de información almacenada en la base de datos.

SQL: Es el lenguaje que se va a utilizar para comunicarnos con la base de datos de la aplicación, pues es aquel que utiliza nuestro sistema gestor de bases de datos: MySQL. Como lenguaje de base de datos que es, nos va a permitir definir el esquema o estructura de nuestra base de datos, y



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

llevar a cabo las operaciones de manipulación necesarias para el correcto funcionamiento del programa.

SQL se va a utilizar de dos formas diferentes para llevar a cabo la creación y modificaciones de la base de datos:

- i. Incorporando las sentencias de código SQL dentro del propio código de la aplicación desarrollado en Python y mediante las funciones que este lenguaje proporciona para el uso de MySQL (mediante el uso del módulo MySQLdb descrito anteriormente en la sección 3.5.2 Librerías de Python).
- ii. Utilizando PHPMyAdmin como herramienta de administración de bases de datos (descrita en la siguiente sección).

Herramienta de administración de bases de datos: PHPMyAdmin

PHPMyAdmin es una herramienta que nos permite administrar bases de datos de una forma bastante sencilla y visual, mucho menos laboriosa que la utilización directa de sentencias SQL. Viene incorporada dentro del paquete de servidor XAMPP. Por lo tanto, no necesita instalación ni configuración añadida. Como su propio nombre indica, está escrita en el lenguaje PHP. Se puede acceder a ella mediante cualquier navegador, pues es una herramienta web. Está alojada dentro del propio servidor web proporcionado por XAMPP, Apache.

PHPMyAdmin también utiliza el lenguaje SQL, solo que lo hace de forma transparente para el usuario. Mediante sus diferentes páginas web se pueden crear, visualizar, modificar y borrar tanto las propias bases de datos como las tablas que estas contienen y sus registros. Entre sus funcionalidades también se encuentran la posibilidad de administrar los usuarios y sus privilegios; hacer copias de seguridad, exportar e importar las bases de datos; e introducir sentencias SQL manualmente.

3.6. Arquitecturas software para el intercambio de información: REST

Para realizar las conexiones de la aplicación con el servidor, y que este pueda proveer los servicios necesarios, debemos fijar una arquitectura software para el intercambio de información entre ambas. Estas arquitecturas son protocolos que definen las pautas de cómo se debe llevar a cabo este traspaso de información. Estas arquitecturas son REST y SOAP [Mayta Flores, C.A. (2012):

SOAP (Simple Object Access Protocol): Es un protocolo para el intercambio de mensajes sobre redes de computadoras, generalmente

CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

usando HTTP. Está basado en XML, esto facilita la lectura, pero también los mensajes resultan más largos y, por lo tanto, considerablemente más lentos de transferir. Utiliza el lenguaje WSDL (Web Services Description Language), el cual nos permite definir claramente cualquier detalle [Mayata Flores, C.A. (2012)].

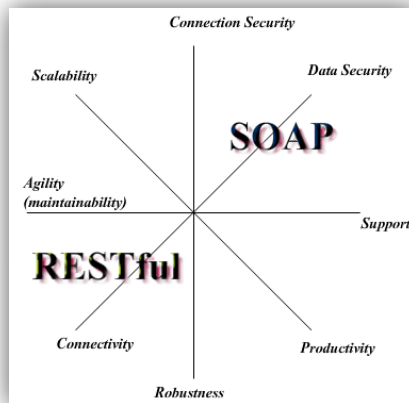


Gráfico 3 – Comparativa de características SOAP y REST

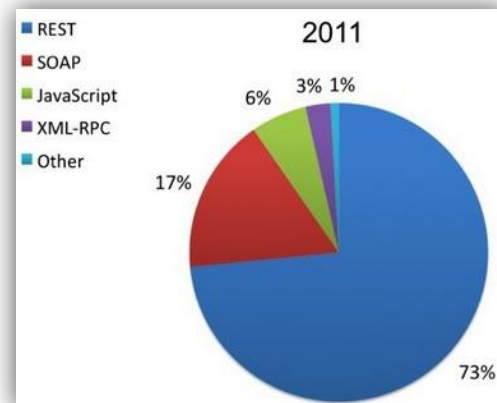


Gráfico 4 – Distribución de protocolos en APIs

En el Gráfico 4 situado a la izquierda se ilustran de forma comparativa las características más importantes de las arquitecturas para el intercambio de información REST y SOAP en 2011 según Andrés Heví.

A la derecha, se sitúa un gráfico sobre la distribución de protocolos para el intercambio de información en APIs en 2011 a nivel mundial, proporcionado por Abel Avram en 2011.

REST (Representational State Transfer): Es un estilo de arquitectura de software para sistemas distribuidos tales como la web, a diferencia de SOAP, se centra en el uso de los estándares HTTP y XML para la transmisión de datos sin la necesidad de contar con una capa adicional. Las operaciones (o funciones) se solicitarán mediante GET, POST, PUT y DELETE, por lo que no requiere de implementaciones especiales para consumir estos servicios. Además se podrá utilizar JSON en vez de XML como contenedor de la información, por lo que será aconsejable utilizar este protocolo cuando busquemos mejorar el rendimiento, o cuando disponemos de escasos recursos, como sería el caso de los dispositivos móviles [Chau, C. (2014)].

SOAP nos ofrece mecanismos de seguridad más eficientes y nos permite abordar funciones más complejas. En consecuencia su desarrollo es más complejo también. En nuestro caso se llevarán a cabo comunicaciones sencillas, y la implementación de una



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

fuerte seguridad no es, al menos por ahora, uno de nuestros objetivos. Por el otro lado REST ofrece un mayor rendimiento, simplicidad, utiliza los estándares y está más extendido. REST permite satisfacer perfectamente las necesidades del proyecto y viene implantado en los módulos de comunicación utilizados.

3.7. Formatos de intercambio de información: JSON

El uso de diferentes lenguajes durante el desarrollo de un proyecto provoca, tarde o temprano, la necesidad de intercambiar datos entre ellos. Para este propósito, surgen estructuras estandarizadas para organizar los datos o información, estos son los formatos de intercambio de información. Los distintos lenguajes han ido adaptándose al uso de estos estándares, permitiendo manejar información estructurada según estos formatos. De esta forma se puede compartir información correctamente estructurada entre ellos de una forma mucho más sencilla.

Los formatos más habituales son XML, JSON y YAML. Cada uno define su propia estructura para organizar la información, así como un tipo de archivo, con extensiones “.xml”, “.json” y “.yaml”, respectivamente. Cualquiera de estos archivos puede ser leído y modificado por un lenguaje de programación capaz de interpretar sus respectivos formatos de intercambio de datos. De esta forma es posible crear un archivo para el intercambio de datos con un lenguaje, y leerlo con otro. También es posible realizar intercambios de información entre lenguajes sin necesidad de estos archivos, y solo mediante funciones propias del lenguaje y la ayuda de otras tecnologías, como por ejemplo AJAX.

XML (Extensible Markup Language): Al igual que HTML, XML es un lenguaje de marcado. Define un conjunto de reglas para codificar información de manera que sea legible por un ser humano y por un ordenador [andrearrr. (2014)]. Esta información codificada se conoce comúnmente como documento, pero gracias a su flexibilidad carece de las limitaciones de los documentos HTML y puede ser usada para representar cualquier estructura de datos.

XML se usa ampliamente para transmitir información en servicios webs y APIs REST y para archivos de configuración. Uno de los lenguajes de programación que le da más soporte es Java.

JSON (JavaScript Object Notation): Es un estándar abierto que utiliza texto plano para codificar información en la forma “*atributo: valor*” [andrearrr. (2014)]. Aunque en sus inicios fue considerado como una



CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS

parte de JavaScript, siempre ha sido independiente del lenguaje de programación y se encuentra disponible para los más populares.

También es ampliamente usado para intercambio de información entre servicios web y APIs REST. Su simplicidad y facilidad de implementación le otorgan un gran desempeño y lo convierten en una de las alternativas ideales al momento de reemplazar XML.

Un objeto JSON es un objeto válido JavaScript por lo que es el formato perfecto para ese lenguaje pero también es empleado con mucha frecuencia por los desarrolladores Python. La mayoría de los navegadores web modernos incluyen funciones nativas para codificar y decodificar JSON, lo que le da un punto de ventaja en lo que se refiere a desempeño y disminuyen los riesgos de seguridad.

YAML (YAML Ain't Another Markup Language): Es otro formato para el intercambio de información que tiene como objetivo facilitar el mapeo de estructuras de datos más complejas (como listas y arreglos asociativos) en un documento de texto plano legible para un ser humano [andearr. (2014)]. Si bien es un formato joven, sus características le han hecho ganarse un lugar importante en la web, junto con XML y JSON.

Es más estricto que los anteriores pero también más simple. Estas características le otorgan elegancia y claridad, haciéndolo ideal para tareas que involucren intervención de un humano.

La simplicidad también le otorga velocidad pero, a diferencia del JSON, no es usado para servicios web o APIs REST sino para archivos de configuración, depuración u otros fines en los que la facilidad de lectura juegan un rol importante. Ha tenido muy buena acogida entre los desarrolladores de Ruby y es ampliamente usado en *frameworks* como Ruby On Rails [andearr. (2014)].

El lenguaje de intercambio de datos JSON será utilizado en el proyecto tanto para guardar archivos “.json” de configuración de la aplicación como para guardar las estructuras de los proyectos de composiciones fotográficas creadas por los usuarios. Su simplicidad, facilidad de implantación y compenetración con Python (a través de la librería “json” incorporada) hacen de esta una opción perfecta para solventar las necesidades de la aplicación.



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica

CAPÍTULO 2: CONOCIMIENTOS TEÓRICOS



CAPITULO 3: DESARROLLO DEL PROYECTO

1. ARCHIVOS BASE

Antes de comenzar con la descripción de algunos de los aspectos técnicos de proyecto, es necesario indicar que la aplicación no parte desde cero. Los recursos iniciales con los que se parte son archivos de tipo imagen, necesarios para crear las composiciones fotográficas de la aplicación. Estos archivos son organizados según la clasificación de elementos necesarios para crear las composiciones: escenarios, personajes y complementos.

Escenarios: Se tienen imágenes relacionadas con los escenarios de tres tipos diferentes:

- i. **Planos de escenario:** Los distintos planos que forman los escenarios están en formato TIFF con una resolución de 3950x2950 y tamaños alrededor de 45 MB cada una. Estas imágenes han sido convertidas a formato PNG (explicado en el apartado 2.1 Tipos de imagen digital del capítulo 2), con resolución aproximada 1920x1440 o 1440x1920 (según la orientación vertical u horizontal) y tamaños entre 1 y 7 MB.
- ii. **Escenarios completos:** Estas son las imágenes de los escenarios sin la división en planos. Disponibles con el mismo formato, resolución y tamaño que los planos. Se convierten a formato PNG con resolución 720x540 o 540x720 con tamaño aproximado de 1 MB.
- iii. **Información sobre escenarios:** Imágenes contienen información sobre la utilización de los escenarios. Están en formato JPEG con resolución 2362x1582 y tamaño aproximado 1 -2 MB.

Personajes: Estas imágenes contienen los personajes disponibles para realizar las composiciones. Están en formato PNG con diferentes resoluciones y tamaños. Se unifican sus resoluciones fijando el alto en 720 pixeles y manteniendo la proporcionalidad de cada una. Sus tamaños oscilan entre los 250 y 1.200 KB.

Complementos: Son los distintos complementos posibles para introducir en las composiciones. También en formato PNG y de diversas resoluciones y tamaños.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

Se unifican fijando el alto a 720 o 360 pixeles según el tamaño del complemento. Se consiguen tamaños de 70 a 1850 KB.

Estas modificaciones de los archivos van a permitir reducir considerablemente el tamaño total de la aplicación, facilitar la manipulación de estas imágenes, mejorar los tiempos de procesamiento y aprovechar las ventajas de los formatos PNG (transparencia) y JPEG (compresión).

2. ESTRUCTURA DE FICHEROS Y DIRECTORIOS

La estructura de directorios y ficheros del proyecto es la siguiente:

Autodesign: Es el directorio raíz del proyecto. Contiene los subdirectorios “config”, “data”, “img” y “video”, los ficheros de programación de Python con extensiones “.py” y “.pyc”, y el manual de usuario en formato PDF:

/config: Este directorio está dedicado a contener los archivos de configuración de la aplicación. Actualmente, solo existe uno:

autodesign.config: Fichero que contiene información acerca de las preferencias de usuario y de la propia aplicación. Su información está en modo texto estructurada según el formato JSON, para poder ser sobrescrita y leída por el programa.

/data: En este directorio se almacenará de forma predeterminada las exportaciones de composiciones fotográficas creadas a través de la aplicación, estas son imágenes en formato PNG o JPG. Además contiene el directorio “saves”:

/saves: En esta carpeta o directorio se alojan, de forma predeterminada, los proyectos de la aplicación que han sido guardados.

Ficheros “.ad”: Son ficheros de extensión propia de la aplicación Autodesign. Son simples archivos de texto que recogen las variables necesarias para la reconstrucción de los proyectos guardados. Estas variables se estructuran también en formato JSON, permitiendo así ser interpretadas con mayor facilidad.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

/img: Directorio que almacena todos los archivos de imagen necesarios para el funcionamiento de la aplicación. En la propia carpeta se almacenan algunas imágenes de tipo genérico y en los subdirectorios imágenes clasificadas de una forma más específica:

/icons: Contiene imágenes en formato PNG que son utilizadas para formar los distintos botones de la aplicación.

/escenarios: Directorio que almacena subdirectorios nombrados de igual forma que los distintos escenarios.

Directorios de escenario: Cada uno de estos directorios se corresponde con uno de los escenarios disponibles y contiene: una imagen del escenario (PNG), una imagen de información del escenario (JPG) y las imágenes correspondientes a los planos del propio escenario: estos tipos están explicados en el apartado anterior.

/personajes: Contiene las imágenes de los personajes en formato PNG y nombradas del mismo modo que en la propia aplicación.

/complementos: Directorio equivalente al anterior pero con los complementos.

/video: Almacena los archivos de vídeo “composcreen1.avi” y “composcreen2.avi”. Estos serán utilizados a modo de “*splash screen*” durante el tiempo inicial de carga del programa.

Ficheros “.py”: Son los archivos propios del lenguaje de programación Python. Recogen el código necesario para el funcionamiento de la aplicación. Cada archivo está dedicado a unas funciones específicas y contiene los módulos de programación dedicados a ellas. Estos ficheros de código se detallan en el siguiente apartado.

Ficheros “.pyc”: Estos ficheros son los archivos compilados de Python. Python es un lenguaje interpretado, lo que significa que cada programa se ejecuta a través del intérprete de Python. Es decir, el intérprete debe “traducir” estos archivos para poder ser ejecutados, lo que implica cierta lentitud. Por ello, se crean automáticamente estos archivos compilados (traducidos) la primera vez que se ejecutan los archivos no compilados. Los archivos “.pyc” no son modificados si no se cambia el

Ejemplo de directorio de uno de los escenarios con sus archivos correspondientes:

Nombre	Tamaño	Dimensiones
Vagón de tren.png	955 KB	723 x 540
Vagón de tren-info.jpg	1.575 KB	2362 x 1582
Vagón de tren-plano1.png	1.681 KB	1930 x 1440
Vagón de tren-plano2.png	1.015 KB	1930 x 1440
Vagón de tren-plano3.png	1.100 KB	1930 x 1440
Vagón de tren-plano4.png	6.552 KB	1930 x 1440
Vagón de tren-plano5.png	3.574 KB	1930 x 1440

Ilustración 12 – Ejemplo de directorio de escenario

Splash screen: Pantalla de bienvenida que se muestra al ejecutar un aplicación software.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

código fuente alojado en los archivos de origen “.py”, de esta forma una vez han sido ejecutados por primera vez no es necesario llevar a cabo una interpretación o traducción para las siguientes. Estos archivos tienen la ventaja de ser más rápidos, y mejorar el tiempo de ejecución.

Autodesign – Manual de Usuario.pdf: Documento en formato PDF que recoge el capítulo 4 de este documento. En éste se explica detalladamente el manejo y funcionamiento de la aplicación de forma orientada a los usuarios.

3. FICHEROS DE CÓDIGO

El código mediante el cual se ejecuta la aplicación es bastante extenso, y es por ello que está organizado siguiendo una lógica. Cada archivo contiene una serie de módulos que producen las distintas funcionalidades del programa. Estos a su vez requieren de otros módulos propios del lenguaje de programación Python. Cada módulo que va a ser utilizado dentro de un archivo y que no está incluido en el mismo debe ser importado. Es posible importar módulos específicos de un archivo “.py” o importar el archivo entero con todos sus módulos. Los ficheros Python desarrollados para la creación de la aplicación son los siguientes:

3.1. `__init__.py`

El archivo `__init__.py` permite declarar como paquete (creado en el proyecto por Eclipse cuando creamos un proyecto) a la carpeta en la que se encuentra. Los archivos suelen estar vacíos y con su simple presencia dan acceso al espacio de nombres.

Estos archivos también permiten disponer de ciertas clases dentro del espacio de nombres. Se ejecutan cada vez que se tiene acceso al paquete y, por lo tanto, cada vez que invocamos el paquete se ejecuta esa fracción de código.

3.2. `main.py`

Éste es el archivo principal del proyecto y, en consecuencia, el más importante. Hace las funciones de archivo raíz dentro del código, a partir de éste se podrá acceder al resto de módulos ubicados en otros archivos. Por ello, este es el archivo procesado para crear la versión del proyecto en forma aplicación, es



CAPÍTULO 3: DESARROLLO DEL PROYECTO

decir, un fichero ejecutable con extensión “.exe”. Todos los proyectos parten de un archivo similar a este.

Su función principal es la de lanzar la aplicación, formando la estructura base de la interfaz gráfica a modo de contenedor donde se sostendrán el resto de elementos. Contiene también los módulos y funciones dedicados a construir los siguientes elementos de la interfaz con sus respectivas funcionalidades:

- i. Panel de inicio
- ii. Panel de registro
- iii. Menú, submenús y barra de herramientas con sus elementos (botones principalmente) y sus respectivas funcionalidades (email, preferencias, cargar, guardar,...).
- iv. Pantalla de bienvenida con vídeo

Importa todos los demás archivos descritos a continuación, pues son necesarios poder crear los distintos paneles, elementos y funcionalidades de la aplicación.

3.3. **panel_usuario.py**

Define la interfaz gráfica del panel de usuario, la cual permite por un lado mostrar y modificar los datos del usuario activo en la aplicación y, por el otro lado, visualizar los proyectos guardados por el usuario para poder cargarlos.

3.4 **panel_imagenes.py**

Se encarga de recoger los datos de los escenarios, personajes y complementos para poder mostrarlos de forma organizada al usuario en panel gráfico de imágenes.

3.5. **panel_edicion.py**

Su función es crear un panel que permite añadir los elementos a la composición, así como editar sus características o eliminarlos. También incorpora las funcionalidades para modificar algunas características fotográficas que afectan de forma global a la composición.

3.6. **panel_proyecto.py**

Se definen las funciones necesarias para cargar las capas de escenario y los elementos seleccionados desde los archivos de imágenes. Realiza también las



CAPÍTULO 3: DESARROLLO DEL PROYECTO

tareas necesarias para crear la composición fotográfica fusionando las distintas capas y aplicando operaciones sobre los archivos de imágenes.

3.7. imagen.py

Define funciones utilizadas por módulos de otros archivos dedicadas a la transformación de imágenes a los distintos formatos que necesita el programa para procesarlas.

3.8. bd.py

Este archivo es el encargado realizar la conexión con la base de datos, definiendo la dirección del servidor donde está alojada y los parámetros necesarios para establecer dicha conexión. Permite realizar las consultas a la base de datos mediante el lenguaje SQL. Se basa en las funciones que incorpora la biblioteca MySQLdb importada.

3.9. config.py

Este fichero permite compartir datos entre los ficheros explicados anteriormente actuando como nexo entre ellos. Es necesario puesto que las variables declaradas como globales en un fichero, están disponibles únicamente para ese fichero. Almacena, durante la ejecución del programa, los datos del proyecto que se está manipulando, el usuario activo y los datos de configuración o preferencias.

3.10. setup.py

El archivo setup.py no se utiliza durante la ejecución del programa, ni aporta funcionalidades al mismo. Su función consiste en generar la aplicación final, agrupando y organizando todos los archivos necesarios, entre ellos un ejecutable, que permitan a los usuarios utilizar la aplicación de forma sencilla. Para ello utiliza la librería py2exe, que permite crear un ejecutable capaz de funcionar en equipos con Windows que no tengan el lenguaje Python instalado.



4. INTERFAZ GRÁFICA DE USUARIO (GUI)

La interfaz gráfica de usuario son el conjunto de formas y métodos que posibilitan al usuario la interacción de forma gráfica con las funcionalidades de la aplicación. No es más que la parte gráfica del programa que el usuario puede ver y controlar, de esta forma puede llevar el control de lo que sucede en la aplicación.

La librería wxPython es la que nos proporciona los recursos necesarios para crear esta interfaz. Permite crear las distintas ventanas de la aplicación, y dividir las en paneles o contenedores en los que se sitúan el resto de elementos gráficos de la interfaz: botones, sliders, imágenes, pestañas,...

La parte gráfica de la aplicación está formada principalmente por tres ventanas: inicio, registro y principal.

4.1. Ventana de inicio

La ventana inicio es la que permite o rechaza la entrada a la aplicación y se puede ver en la *Ilustración 14*. Para ello solicita los datos de entrada al usuario (usuario y clave), los coteja con la base de datos, y si son correctos nos permite el paso a la ventana principal del programa. Durante la transición de estas ventanas se lanza una ventana que contiene el video que actúa como pantalla de bienvenida (*Ilustración 15*) durante el periodo de carga de la aplicación. También nos permite abrir la ventana de registro.

CAPÍTULO 3: DESARROLLO DEL PROYECTO

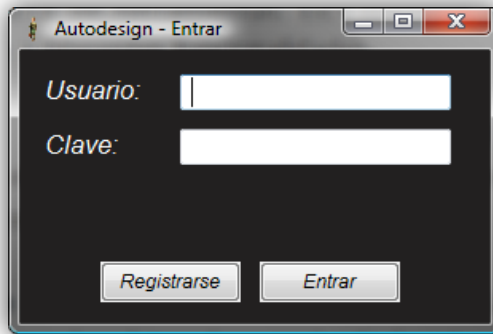


Ilustración 14 – Vista de la ventana de inicio



Ilustración 15 - Vista de la ventana de bienvenida



Ilustración 13 – Vista de la ventana de registro

4.2. Ventana de registro

Muestra una serie de campos de texto que el usuario debe rellenar correctamente para darse de alta en la aplicación, como se puede observar en la *Ilustración 13*. Una vez se ha validado el formato de los datos introducidos, el registro es correcto y se lanza automáticamente la ventana de inicio.

4.3. Ventana principal

Como su propio nombre indica, es la ventana principal de la aplicación y contiene prácticamente todos los recursos gráficos, a excepción de las funcionalidades accesibles mediante las dos ventanas anteriormente mencionadas. Permite además, mostrar otras ventanas secundarias, como se aprecia en la *Ilustración 21*. En la parte superior se dispone el menú, en el cual se despliegan los submenús que ofrecen las distintas funcionalidades que ofrece la aplicación. En la *Ilustración 22* se puede ver barra de herramientas que se encuentra bajo el

CAPÍTULO 3: DESARROLLO DEL PROYECTO

menú. Ésta contiene iconos que actúan como accesos directos a diferentes funcionalidades.

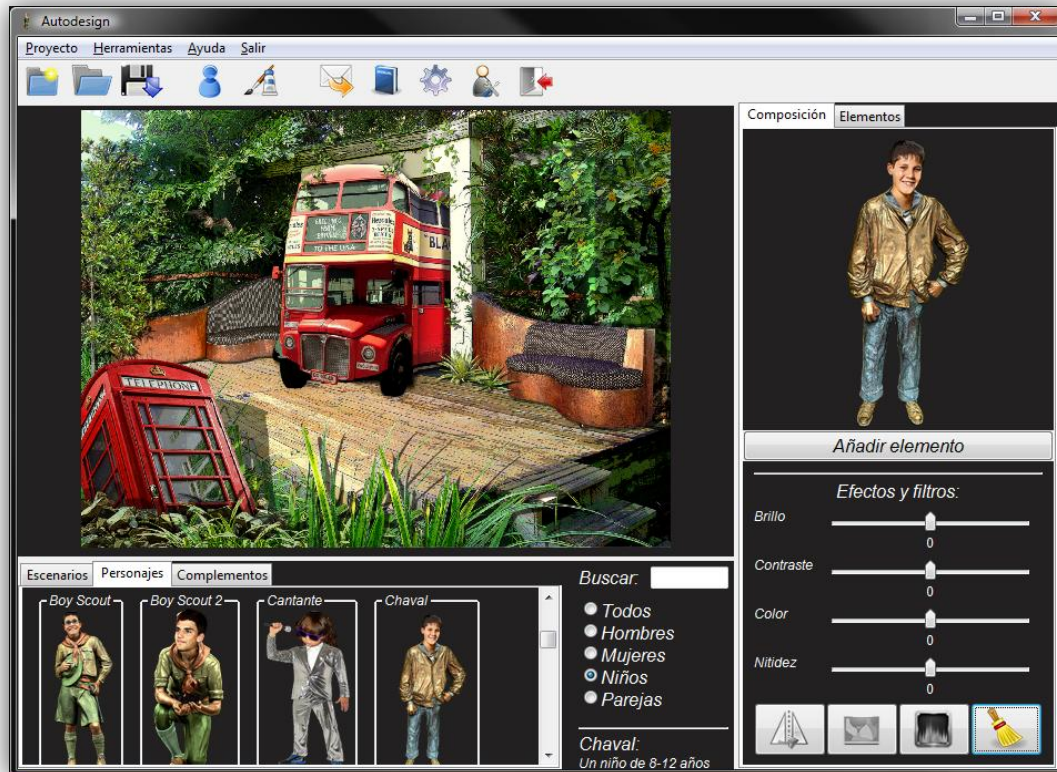


Ilustración 16 – Vista de la ventana principal



Ilustración 17 – Vista de la barra de herramientas

Bajo la barra de herramientas se encuentra el panel de edición de composiciones, el cual se puede conmutar por el panel de usuario. Estos dos paneles tienen como función contener otros paneles.

Panel de usuario: Se divide en dos secciones. La sección de la izquierda nos muestra los datos de usuario de la misma forma que el registro, permitiéndonos modificarlos. Nos ofrece la opción de cambiar la clave del usuario, para lo que se desplegará una nueva ventana. La sección derecha del panel de usuario nos proporciona los proyectos guardados por el usuario activo en la aplicación, mostrando la composición fotográfica de cada proyecto en el momento de su guardado.



CAPÍTULO 3: DESARROLLO DEL PROYECTO



Ilustración 18 – Vista del panel de usuario

Panel de composición: Está dividido en diferentes partes:

Panel de imágenes: Se muestra en la parte inferior de la aplicación. Está compuesto por tres paneles más en forma de páginas intercambiables por pestañas. Estas páginas contienen las miniaturas de escenarios, personajes y complementos. Contiene también un buscador para estos elementos y panel informativo.

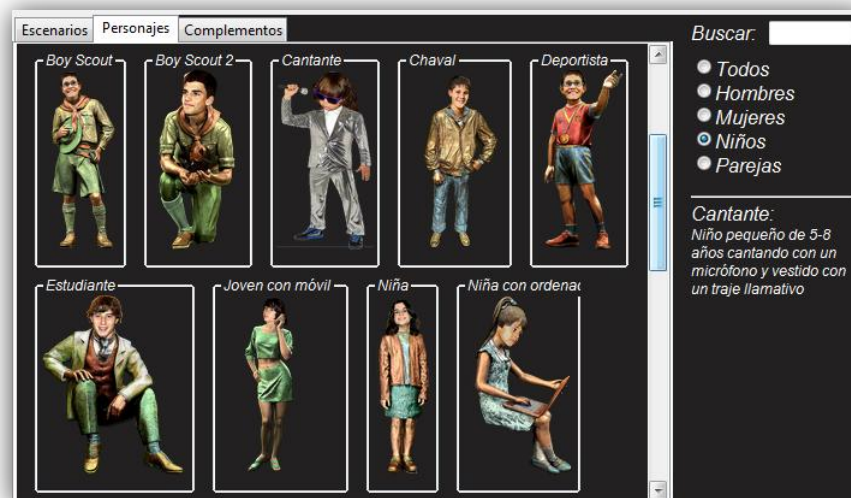


Ilustración 19 – Vista del panel de imágenes

CAPÍTULO 3: DESARROLLO DEL PROYECTO

Panel de edición: Colocado a la derecha del panel de composiciones y estructurado también con páginas intercambiables por pestañas. La página de composición permite añadir un nuevo elemento a la composición, mostrándolo previamente, y ajustar los efectos y filtros disponibles. La página de elementos nos muestra los elementos que ya forman parte de la composición y sus características. Estas características se muestran en controles y botones, permitiendo al usuario realizar ajustes sobre ellas.

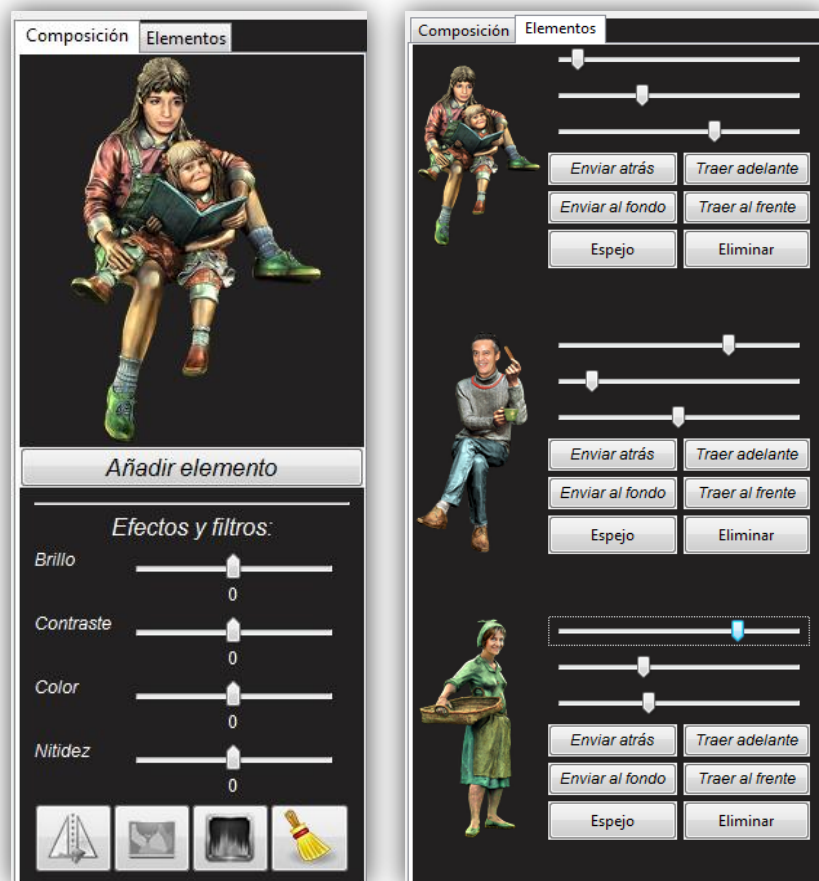


Ilustración 20 – Vistas del panel de edición

Panel de proyecto: Este panel se sitúa sobre la parte superior izquierda del panel de composición. Muestra gráficamente el estado actual de la composición fotográfica escalando la imagen según el espacio disponible para este panel. En el podemos ver el



CAPÍTULO 3: DESARROLLO DEL PROYECTO

resultado provocado por los cambios y ajustes realizados mediante los paneles mencionados anteriormente.

Mediante los elementos de los submenús o de la barra de herramientas se pueden ejecutar funcionalidades que hacen uso de ventanas gráficas secundarias. Las funciones de guardar, cargar y exportar proyectos muestran ventanas de selección de archivos y formatos. Mientras que las funcionalidades de información de escenario, enviar un correo, preferencias, cambiar clave e información hacen visibles distintas ventanas personalizadas para dichas funciones. En las siguientes ilustraciones se pueden ver algunas de estas ventanas:

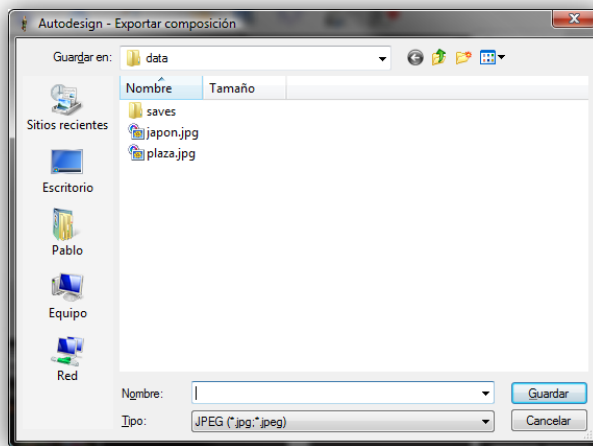


Ilustración 21 – Vista de la ventana de exportación

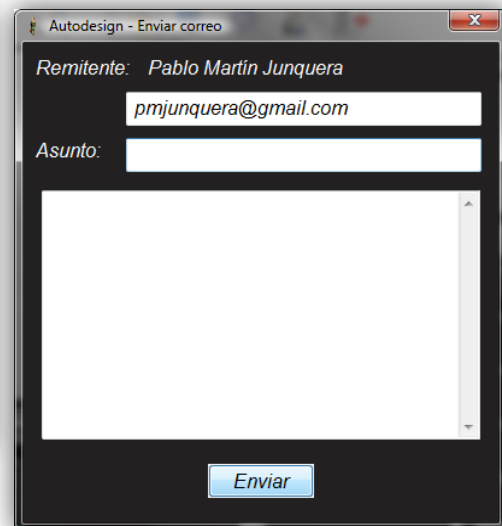


Ilustración 22 – Vista de la ventana de enviar correo

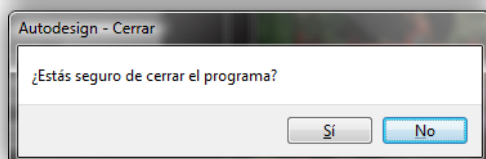
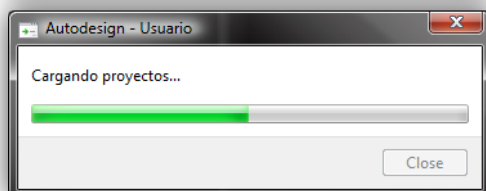
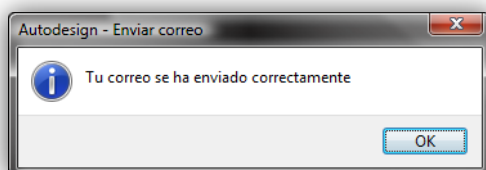


Ilustración 24 – Vistas de ventanas informativas y de control

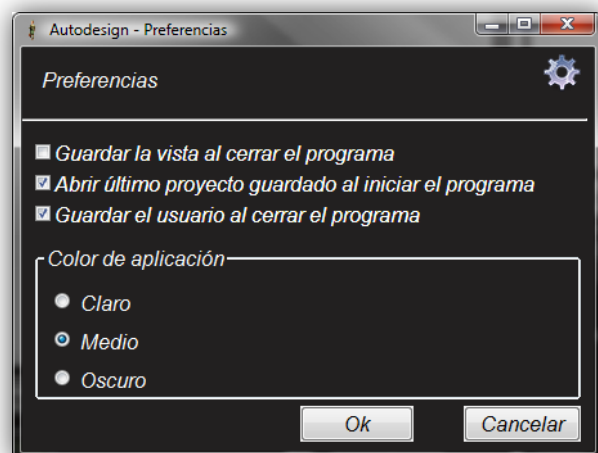


Ilustración 23 – Vista de la ventana de preferencias



CAPÍTULO 3: DESARROLLO DEL PROYECTO

4.4. Otras ventanas

Otras ventanas no mencionadas pueden aparecer durante la ejecución de una instancia del programa. Estas son utilizadas para mostrar información, alertas, errores, preguntas, barras de carga,... Este tipo de ventanas mejoran la usabilidad de y permiten controlar el flujo de la aplicación en cierta medida, pero no ofrecen funcionalidades añadidas.

5. BASE DE DATOS

La aplicación hace uso de información que ha de ser almacenada y organizada correctamente. Para ello se hace so, como ya se ha mencionado anteriormente, de una base de datos. La información en la base de datos está perfectamente organizada para facilitar el uso de esta por parte la aplicación.

En la base de datos se almacena tanto datos utilizados por el propio programa para llevar a cabo unas determinadas tareas, como información que ha de ser presentada a los usuarios de la aplicación.

A continuación, se detallan las tablas que componen la estructura de la base de datos, para comprender así su funcionamiento y utilidad.

5.1. Tabla usuarios

La tabla *usuarios* se utiliza para almacenar los datos personales de los usuarios que se hayan registrado correctamente en la aplicación. Solo pueden hacer uso de la aplicación aquellas personas que tengan un registro en esta tabla. De esta forma, los registros de esta tabla crecerán a medida que los usuarios vayan haciendo uso de la aplicación. Algunos campos simplemente almacenan información del usuario en cuestión, sin embargo, otros se utilizan en distintas funcionalidades del programa.

usuario: Es el nombre que recibe un usuario. Este campo constituye la clave primaria de la tabla. Está en formato VARCHAR y permite 12 caracteres. No puede haber más de un usuario con el mismo nombre de usuario. Cuando se realiza el registro en la aplicación, si se introduce un nombre de usuario ya existente se notifica al usuario para que elija otro

CAPÍTULO 3: DESARROLLO DEL PROYECTO

distinto. Los proyectos guardados se asignan también a un usuario, guardando consigo este nombre de usuario.

clave: Campo de tipo VARCHAR con longitud de 12 caracteres. Es la contraseña personal de cada usuario. Permite identificar de forma segura a cada persona con el usuario de la aplicación. Junto con el campo *usuario* son los datos solicitados en la ventana de inicio para poder acceder a la aplicación. Se permite a los usuarios cambiar de clave desde la opción mostrada en el panel de usuarios.

Campos
tipo CHAR(13)
dni CHAR(9)
nombre CHAR(30)
apellidos CHAR(30)
email CHAR(50)
usuario CHAR(12)
clave CHAR(12)
telefono CHAR(9)
direccion CHAR(30)
provincia CHAR(30)
localidad CHAR(30)
cp CHAR(5)

Ilustración 25 – Tabla usuarios

Los campos *dni*, *nombre*, *apellidos*, *telefono*, *direccion*, *provincia*, *localidad* y *cp* son, por el momento, campos que simplemente almacenan información acerca del usuario. Pero puede que sean utilizados en futuras incorporaciones de funcionalidades.

Todos los campos de la tabla deben cumplir una serie de características en cuanto a formato antes de ser introducidos en la base de datos. La comprobación de estas características se realiza durante la ejecución de la aplicación, informando al usuario de cómo debe introducir estos datos

tipo: Este campo es también de tipo VARCHAR y permite una longitud de 13 caracteres. Los tipos manejados por la aplicación son “administrador” y “cliente”. Los usuarios registrados mediante la ventana de registro de la aplicación tienen este campo asignado como “cliente”. Los usuarios que tienen este campo asignado como “administrador” son introducidos manualmente en la base de datos. Estos últimos son aquellos que tendrán permisos de administrador. Sólo estos últimos pueden acceder a la funcionalidad de administrador de la aplicación, ya que a los de tipo “cliente” se les denegará el acceso al área de administración.

email: Campo de tipo VARCHAR que permite cadenas de hasta 50 caracteres. Almacena el correo introducido por el usuario, el cual será utilizado de forma predeterminada como campo remitente en la funcionalidad de “enviar correo”.

CAPÍTULO 3: DESARROLLO DEL PROYECTO

correctamente. Como ya se ha mencionado anteriormente, los datos en esta tabla son introducidos mediante el formulario de registro, pero cada usuario puede modificar los datos de su propio registro mediante el panel de usuario disponible en la aplicación.

5.2. Tablas de escenarios, personajes y complementos

Las tablas nombradas como *escenarios*, *personajes* y *complementos* son utilizadas para almacenar información acerca de los elementos que forman parte de las composiciones fotográficas. Estos son, como los propios nombres de las tablas indican: escenarios, personajes y complementos. Se almacenan tanto datos de carácter funcional, que permiten al programa realizar sus tareas, como datos de carácter informativo que serán mostrados a los usuarios.

escenarios
id_escenario INT
nombre_escenario CHAR(30)
descripcion TEXT
ruta_escenario CHAR(100)
Indexes

Ilustración 28 – Tabla escenarios

personajes
id_personaje INT
nombre_personaje CHAR(30)
tipo CHAR(1)
descripcion TEXT
ruta_personaje CHAR(100)
Indexes

Ilustración 26 – Tabla personajes

complementos
id_complemento INT
nombre_complemento CHAR(30)
descripcion TEXT
ruta_complemento CHAR(100)
Indexes

Ilustración 27 – Tabla complementos

Los campos disponibles en estas tablas son los siguientes:

id_escenario, id_personaje, id_complemento: Son campos de tipo INT, auto incrementales y son las claves primarias de sus tablas correspondientes. Son los identificadores de los posibles elementos de una composición. Mediante estos campos se puede identificar de forma inequívoca a los distintos escenarios, personajes o complementos, ya que al ser un campo definido como clave primaria, este no puede estar repetido.

nombre_escenario, nombre_personaje, nombre_complemento: Campos de tipo VARCHAR que almacenan el nombre dado a cada elemento. Se utilizan en la aplicación para buscar determinados archivos locales de imagen con el mismo nombre y para mostrar al usuario el nombre que recibe cada escenario, personaje o complemento. En la funcionalidad de buscar, también es utilizado puesto que la búsqueda de coincidencias con la cadena de caracteres introducida por un usuario se realiza, entre otros, en este campo de los registros.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

descripcion: Este campo aparece en las tres tablas vistas en este apartado. Es de tipo TEXT y por lo tanto almacena cadenas de caracteres sin límite. En ellos se almacena una breve descripción sobre cada uno de los elementos fotográficos, dependiendo de la tabla. El campo *descripcion* se utiliza de la misma forma que los campos de nombre en la funcionalidad de búsqueda. De esta forma, las búsquedas realizadas por los usuarios se realizarán tanto en los campos que almacenan nombres como en los que almacenan descripciones. Las descripciones guardadas en estos campos también serán mostradas a los usuarios dentro del panel de imágenes de la aplicación.

tipo: Es un campo de tipo CHAR que permite almacenar un único carácter de texto. Está disponible solo en la tabla *personajes* puesto que se utiliza para diferenciar los distintos tipos de personajes. Estos son: hombre (h), mujer (m), niño (n) y pareja (p). De esta forma los usuarios de la aplicación también pueden buscar o clasificar los personajes de acuerdo según esta característica. Esta funcionalidad también está disponible dentro del panel de imágenes.

Los registros de estas tablas son introducidos manualmente en la base de datos, a diferencia del caso anterior, en el que eran los usuarios los que actualizaban la lista de registros. Se pretende crear un área de administrador que permita los usuarios del tipo “Administrador” añadir, modificar o eliminar registros en estas tablas. Esta idea se explica posteriormente en el apartado de “líneas futuras”.

6. FUNCIONALIDADES

En este apartado se describe el funcionamiento de la aplicación. Para hacerlo de una forma clara y organizada se describen por separado cada una de sus funcionalidades. Las funcionalidades son descritas mediante casos de uso recogidos en forma de tabla.

6.1. Registrarse

Caso de Uso	CU1: Registrarse
Actor	Cliente, Administrador
Descripción	El usuario puede darse de alta en la aplicación rellenando correctamente un formulario de registro.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Ejecutar la aplicación. 2. Presionar el botón “registrarse” en la ventana de inicio. 3. Rellenar el formulario de registro correctamente. 4. Presionar el botón “registrarse” en la venta de registro. 5. Presionar el botón “OK” para confirmar el registro.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

Flujos alternativos (FA)	<p>FA1. Datos de entrada incorrectos En el paso 4 del FB la aplicación detecta que el usuario ha introducido alguno de los campos del formulario de forma incorrecta. Se vuelve a FB.3 (Paso 3 del flujo básico). Podrá volver a modificar cualquiera de los campos hasta conseguir rellenar todos de forma correcta.</p> <p>FA2. Cancelar el registro Durante el FB.3 el usuario decide cancelar el registro presionando el botón “Volver” de la ventana de registro. El flujo finaliza cumpliendo solo la segunda post-condición.</p>
Pre-condiciones	Ninguna.
Post-condiciones	<ol style="list-style-type: none"> 1. Los datos de usuario son introducidos correctamente en la base de datos. 2. La ventana de registro se cierra y la ventana de inicio se abre.
Requerimientos	<ol style="list-style-type: none"> 1. Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos.
Comentario	<ol style="list-style-type: none"> 1. Si uno de los campos introducidos por el usuario no cumple las características formato necesarias se muestra una nota que indica cómo ha de introducirse dicho dato, el campo se vuelve de color rojo, y el cursor de escritura se sitúa sobre dicho campo. 2. Una vez completado el registro el usuario puede entrar en la aplicación mediante el caso de uso “Entrar”.

Tabla 4 - Caso de uso: Registrarse

6.2. Entrar

Caso de Uso	CU2: Entrar
Actor	Cliente, Administrador
Descripción	El usuario puede entrar en la aplicación introduciendo su nombre de usuario y clave.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Ejecutar la aplicación. 2. Rellenar los campos de usuario y clave de la ventana de inicio. 3. Presionar el botón “Entrar”.
Flujos alternativos (FA)	<p>FA1. Datos de entrada incorrectos En FB.3 la aplicación detecta que el usuario ha introducido alguno de los campos del formulario de forma incorrecta. Se vuelve a FB.2. El usuario debe volver a introducir los datos de forma correcta.</p> <p>FA2. Datos de entrada recordados En caso de que se haya activado la opción “Guardar el usuario al cerrar el programa” en las preferencias (CU7), si el usuario lo desea puede omitir el paso FB.2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe haber completado con anterioridad (en cualquier ejecución del programa) el proceso de registro completo, mediante el CU2 (“Caso de Uso 2: Registrarse”).
Post-condiciones	<ol style="list-style-type: none"> 1. La ventana de inicio se cierra.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	<ol style="list-style-type: none"> La ventana de bienvenida de abre (<i>splashscreen</i> en forma de vídeo) durante el período de carga. Tras la carga completa de la aplicación la ventana principal se abre con la sesión del correspondiente al nuevo usuario introducido. Si se ha activado anteriormente la opción “Guardar la vista al cerrar el programa” en las preferencias (CU7), la ventana principal se cargará con el aspecto que tenía en el momento de cierre de la última ejecución. En caso de que se haya activado la opción “Abrir último proyecto guardado al iniciar el programa” en las preferencias (CU7), se cargará el último proyecto guardado si pertenece al usuario que ha iniciado sesión.
Requerimientos	<ol style="list-style-type: none"> Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos.
Comentario	<ol style="list-style-type: none"> La aplicación coteja los datos de usuario y clave con la base de datos. Si alguno de los datos introducidos no se corresponde se muestra una nota indicando al usuario el problema, el campo se vuelve de color rojo, y el cursor de escritura se sitúa sobre dicho campo.

Tabla 5– Caso de uso: Entrar

6.3. Modificar los datos de usuario

Caso de Uso	CU3: Modificar los datos de usuario
Actor	Cliente, Administrador
Descripción	El usuario puede modificar los datos de su registro reescribiendo un formulario rellenado previamente con sus datos de registro actuales.
Flujo básico (FB)	<ol style="list-style-type: none"> Acceder al panel de usuario mediante el menú o la barra de herramientas. Rellenar el formulario de datos de usuario correctamente. Presionar el botón “Actualizar datos” en el panel de usuario. Presionar el botón “OK” para confirmar los cambios.
Flujos alternativos (FA)	<p>FA1. Vista del panel de usuario En caso de estar visible el panel de usuario el flujo comienza en FB.2.</p> <p>FA2. Datos de entrada incorrectos En el paso 3 del FB la aplicación detecta que el usuario ha introducido alguno de los campos del formulario de forma incorrecta. Se vuelve a FB.2. El usuario podrá volver a modificar cualquiera de los campos hasta conseguir rellenar todos de forma correcta.</p>
Pre-condiciones	<ol style="list-style-type: none"> El usuario debe haber completado con anterioridad (en cualquier ejecución del programa) el proceso de registro completo, mediante el CU (caso de uso) “Registrarse”. Entrada satisfactoria en la aplicación (CU2)
Post-condiciones	<ol style="list-style-type: none"> Los datos de usuario son modificados correctamente en la base



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	de datos.
Requerimientos	1. Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos.
Comentario	1. Si uno de los campos introducidos por el usuario no cumple las características formato necesarias se muestra una nota que indica cómo ha de introducirse dicho dato, el campo se vuelve de color rojo, y el cursor de escritura se sitúa sobre dicho campo.

Tabla 6 – Caso de uso: Modificar los datos de usuario

6.4. Cambiar la clave de usuario

Caso de Uso	CU4: Cambiar clave de usuario
Actor	Cliente, Administrador
Descripción	El usuario puede cambiar su clave de usuario.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de usuario mediante el menú o la barra de herramientas. 2. Presionar el botón “Cambiar clave” del panel de usuario. 3. Rellenar el formulario de datos con la clave anterior y la clave nueva dos veces. 4. Presionar el botón “Cambiar clave” en la misma ventana. 5. Presionar el botón “OK” para confirmar los cambios.
Flujos alternativos (FA)	<p>FA1. Vista del panel de usuario En caso de estar visible el panel de usuario el flujo comienza en FB.2.</p> <p>FA2. Datos de entrada incorrectos En FB.4 la aplicación coteja los datos introducidos con la base de y detecta que el usuario ha introducido alguno de los campos del formulario de forma incorrecta. Se vuelve a FB.3. El usuario debe volver a introducir los datos de forma correcta.</p> <p>FA3. Cerrar la ventana de cambiar clave Durante el FB.3 el usuario decide cerrar la ventana de cambiar clave. El flujo finaliza inmediatamente sin cumplir las post-condiciones.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe haber completado con anterioridad (en cualquier ejecución del programa) el proceso de registro completo, mediante el CU (caso de uso) “Registrarse”. 2. Entrada satisfactoria en la aplicación (CU2) 3. El usuario debe recordar su antigua clave para poder cambiarla por una nueva.
Post-condiciones	1. La clave de usuario es modificada correctamente en la base de datos.
Requerimientos	1. Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos.
Comentario	1. La aplicación coteja la clave antigua con la base de datos. Si la clave introducida no se corresponde se muestra una nota indicando al usuario el problema, el campo se vuelve de color



CAPÍTULO 3: DESARROLLO DEL PROYECTO

rojo, y el cursor de escritura se sitúa sobre dicho campo.
--

Tabla 7 – Caso de uso: Cambiar clave de usuario

6.5. Enviar un correo

Caso de Uso	CU5: Enviar un correo
Actor	Cliente, Administrador
Descripción	El usuario puede enviar un correo para notificar alguna incidencia o preguntar sobre la aplicación a una cuenta asociada al administrador de la aplicación.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder a la ventana de enviar correo mediante el menú o la barra de herramientas. 2. Rellenar el formulario de datos con el correo remitente, asunto y mensaje. 3. Presionar el botón “Enviar” en la ventana de enviar correo. 4. Presionar el botón “OK” para confirmar el envío de mensaje correcto.
Flujos alternativos (FA)	<p>FA1. Datos de entrada no introducidos En FB.3 si los campos remitente o asunto están vacíos en envío del correo no se efectuará. Se vuelve a FB.2.</p> <p>FA2. Error en el envío En FB.3, en caso de producirse un error durante el envío del correo se abrirá una ventana de alerta indicando el error al usuario. El flujo finaliza inmediatamente sin cumplir las post-condiciones.</p> <p>FA3. Cerrar la ventana de enviar correo Durante el FB.2 el usuario decide cerrar la ventana de cambiar de usuario. El flujo finaliza inmediatamente sin cumplir las post-condiciones.</p>
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	<ol style="list-style-type: none"> 1. La ventana de enviar correo se cierra. 2. Se envía un correo al administrador de la aplicación con los datos introducidos en el formulario.
Requerimientos	1. Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos y con el servidor de correo.
Comentario	1. En FA1, el campo que se encuentra vacío se vuelve de color rojo, y el cursor de escritura se sitúa sobre dicho campo.

Tabla 8 – Caso de uso: Enviar un correo



CAPÍTULO 3: DESARROLLO DEL PROYECTO

6.6. Acceder al área de administración

Caso de Uso	CU6: Acceder al área de administración
Actor	Administrador
Descripción	El usuario puede acceder al área de administración de la aplicación desde la misma de forma directa.
Flujo básico (FB)	1. Acceder al área de administración mediante el menú o la barra de herramientas.
Flujos alternativos (FA)	FA1. Usuario no autorizado El usuario que ha iniciado sesión en la aplicación no tiene permisos de administrador. Una ventana de alerta se abre explicando al usuario este hecho. El flujo termina finaliza sin cumplir las post-condiciones.
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2). 2. El usuario debe estar registrado en la base de datos como "administrador".
Post-condiciones	1. Se abre un navegador de Internet con el área de administrador como sitio web.
Requerimientos	1. Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos y con el área de administración web. 2. Es necesario tener instalado en la computadora un navegador de Internet.
Comentario	1. El área de administración permite a un usuario administrador gestionar la base de datos de la aplicación.

Tabla 9 – Caso de uso: Acceder al área de administración

6.7. Editar las preferencias

Caso de Uso	CU7: Editar las preferencias
Actor	Cliente, Administrador
Descripción	El usuario puede modificar algunas de las configuraciones de usuario y aplicación disponibles.
Flujo básico (FB)	1. Acceder a la ventana de preferencias mediante el menú o la barra de herramientas. 2. Activar o desactivar las casillas de las distintas preferencias de usuario. 3. Elegir mediante los botones una tonalidad de color para el fondo de la aplicación. 4. Presionar el botón "OK" del panel de preferencias para confirmar los cambios en la configuración.
Flujos alternativos (FA)	FA1. Cerrar la ventana de preferencias o presionar el botón "Cancelar". Durante el FB.2 o el FB.3 el usuario decide cerrar la ventana de preferencias o presionar el botón "cancelar". El flujo finaliza inmediatamente sin cumplir las post-condiciones.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	1. La ventana de preferencias se cierra. 2. Las preferencias son guardadas en el archivo de configuración.
Requerimientos	Ninguno
Comentario	1. Activar la casilla “Guardar la vista al cerrar el programa” implica que cuando el programa se cierre guardara la posición y tamaño de la ventana principal y de sus paneles. Cuando la aplicación vuelva a ser ejecutada, cargará la vista guardada. 2. Activar la casilla “Abrir último proyecto guardado al iniciar el programa” permite a la aplicación recordar cuál fue el último proyecto guardado y abrirlo directamente entrar en el programa la siguiente vez. 3. Activar la casilla “Guardar el usuario al cerrar el programa” permite a la aplicación recordar los datos de usuario y contraseña del último usuario que inició sesión al ejecutar de nuevo la aplicación. 4. Seleccionar un color de aplicación permite ajustar el color de fondo de la aplicación entre tres disponibles, pero los cambios estéticos afectan a partir de la siguiente ejecución del programa.

Tabla 10 – Caso de uso: Editar las preferencias

6.8. Mostrar información del programa

Caso de Uso	CU8: Mostrar la información del programa
Actor	Cliente, Administrador
Descripción	El usuario puede consultar información acerca de la aplicación que está utilizando.
Flujo básico (FB)	1. Acceder a la ventana de información mediante el menú o la barra de herramientas. 2. Presionar en los distintos botones de la ventana de información para mostrar distintos tipos de información acerca de la aplicación.
Flujos alternativos (FA)	Ninguno
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	Ninguna
Requerimientos	Ninguno
Comentario	Ninguno

Tabla 11 – Caso de uso: Mostrar información del programa



CAPÍTULO 3: DESARROLLO DEL PROYECTO

6.9. Abrir el manual de usuario

Caso de Uso	CU9: Abrir el manual de usuario
Actor	Cliente, Administrador
Descripción	El usuario puede consultar el manual de usuario de la aplicación accediendo directamente desde la aplicación.
Flujo básico (FB)	1. Acceder al manual de usuario mediante el menú o la barra de herramientas.
Flujos alternativos (FA)	Ninguno
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	Ninguna
Requerimientos	Disponer de un programa instalado capaz de leer documentos en formato PDF.
Comentario	El manual de usuario se abre en formato PDF mediante un programa externo.

Tabla 12– Caso de uso: Abrir el manual de usuario

6.10. Buscar escenarios

Caso de Uso	CU10: Buscar escenarios
Actor	Cliente, Administrador
Descripción	El usuario puede realizar búsquedas de escenarios entre los disponibles en la aplicación, mostrándose solo aquellos que coincidan con los parámetros de búsqueda.
Flujo básico (FB)	1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Escenarios” dentro del panel de imágenes para mostrar la página de escenarios. 3. Introducir caracteres de texto en la casilla de texto situada a la derecha de “Buscar” dentro del panel de imágenes.
Flujos alternativos (FA)	FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2. FA2. Vista de la página de escenarios En caso de estar visible la página de escenarios el flujo comienza en FB.3.
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	1. Los escenarios mostrados en la página de escenarios cambian de acuerdo a la búsqueda realizada por el usuario.
Puntos de extensión	E1. Descripción de escenario A partir de FB.3 es posible seleccionar uno de los escenarios haciendo “click” sobre uno de ellos. Esto permite mostrar el nombre del escenario, su descripción y el botón información en el panel de imágenes.
Requerimientos	1. Es necesaria conexión a Internet para comunicarse con el



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	servidor que contiene la base de datos.
Comentario	<ol style="list-style-type: none"> 1. Para mostrar cada uno de los escenarios, se muestra una imagen en miniatura del propio escenario y su correspondiente nombre. 2. Por defecto, si no se introduce ningún patrón de búsqueda, se muestran todos los escenarios disponibles.

Tabla 13– Caso de uso: Buscar escenarios

6.11. Mostrar información de escenario

Caso de Uso	CU11: Mostrar información de escenario
Actor	Cliente, Administrador
Descripción	El usuario puede consultar información añadida sobre cada escenario. Se muestran los distintos planos de cada escenario y se amplía la información para su uso.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Escenarios” dentro del panel de imágenes para mostrar la página de escenarios. 3. Seleccionar uno de los escenarios visibles dentro de la página de escenarios mediante un “click”. 4. Pulsar el botón “información” para abrir la ventana de información de escenario.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de escenarios En caso de estar visible la página de escenarios el flujo comienza en FB.3.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2). 2. Al menos un escenario visible en la página de escenarios.
Post-condiciones	<ol style="list-style-type: none"> 1. La ventana de información de escenario se abre.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. Los escenarios visibles en la página de escenarios dependerán de la búsqueda realizada. Ver CU10. 2. La información de escenario es mostrada en formato de imagen sobre la ventana.

Tabla 14– Caso de uso: Mostrar información de escenario

6.12. Cargar escenario

Caso de Uso	CU12: Cargar escenario
Actor	Cliente, Administrador
Descripción	El usuario puede cargar uno de los escenarios disponibles en la composición.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	<p>herramientas.</p> <ol style="list-style-type: none"> 2. Seleccionar la pestaña “Escenarios” dentro del panel de imágenes para mostrar la página de escenarios. 3. Seleccionar uno de los escenarios visibles dentro de la página de escenarios mediante un “doble click”.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de escenarios En caso de estar visible la página de escenarios el flujo comienza en FB.3.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2). 2. Al menos un escenario visible en la página de escenarios.
Post-condiciones	<ol style="list-style-type: none"> 1. El escenario aparece en la composición ubicada en el panel proyecto.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. Los escenarios visibles en la página de escenarios dependerán de la búsqueda realizada. Ver CU10.

Tabla 15 – Caso de uso: Cargar escenario

6.13. Buscar personajes

Caso de Uso	CU13: Buscar personajes
Actor	Cliente, Administrador
Descripción	El usuario puede realizar búsquedas de personajes entre los disponibles en la aplicación, mostrándose solo aquellos que coincidan con los parámetros de búsqueda.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Personajes” dentro del panel de imágenes para mostrar la página de escenarios. 3. Introducir caracteres de texto en la casilla de texto situada a la derecha de “Buscar” dentro del panel de imágenes. 4. Seleccionar el tipo de personaje mediante los botones de tipo radio disponibles en el panel de imágenes.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de escenarios En caso de estar visible la página de personajes el flujo comienza en FB.3.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2)
Post-condiciones	<ol style="list-style-type: none"> 1. Los personajes mostrados en la página de personajes cambian de acuerdo a la búsqueda realizada y al tipo de personaje seleccionado por el usuario.
Puntos de	E1. Descripción de personaje



CAPÍTULO 3: DESARROLLO DEL PROYECTO

extensión	A partir de FB.3 es posible seleccionar uno de los personajes haciendo “click” sobre uno de ellos. Esto permite mostrar el nombre del personaje y su descripción en el panel de imágenes.
Requerimientos	1. Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos.
Comentario	1. Para mostrar cada uno de los personajes, se muestra una imagen en miniatura del propio personaje y su correspondiente nombre. 2. Por defecto, si no se introduce ningún patrón de búsqueda, se muestran todos los personajes disponibles. 3. Las opciones para seleccionar el tipo de personaje son: “Todos”, “Hombres”, “Mujeres”, “Niños” y “Parejas”.

Tabla 16 – Caso de uso: Buscar personajes

6.14. Cargar personaje

Caso de Uso	CU14: Cargar personaje
Actor	Cliente, Administrador
Descripción	El usuario puede cargar uno de los personajes disponibles en el panel de edición para verlo con mayor detalle.
Flujo básico (FB)	1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Personajes” dentro del panel de imágenes para mostrar la página de personajes. 3. Seleccionar uno de los personajes visibles dentro de la página de personajes mediante un “doble click”.
Flujos alternativos (FA)	FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2. FA2. Vista de la página de personajes En caso de estar visible la página de personajes el flujo comienza en FB.3.
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2). 2. Al menos un personaje visible en la página de personajes.
Post-condiciones	1. El personaje se muestra en el panel de edición.
Requerimientos	Ninguno
Comentario	1. Los personajes visibles en la página de personajes dependerán de la búsqueda realizada. Ver CU13.

Tabla 17 – Caso de uso: Cargar personaje



CAPÍTULO 3: DESARROLLO DEL PROYECTO

6.15. Buscar complementos

Caso de Uso	CU15: Buscar complementos
Actor	Cliente, Administrador
Descripción	El usuario puede realizar búsquedas de complementos entre los disponibles en la aplicación, mostrándose solo aquellos que coincidan con los parámetros de búsqueda.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Complementos” dentro del panel de imágenes para mostrar la página de complementos. 3. Introducir caracteres de texto en la casilla de texto situada a la derecha de “Buscar” dentro del panel de imágenes.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de complementos En caso de estar visible la página de complementos el flujo comienza en FB.3.</p>
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2)
Post-condiciones	1. Los complementos mostrados en la página de complementos cambian de acuerdo a la búsqueda realizada por el usuario.
Puntos de extensión	<p>E1. Descripción de complemento A partir de FB.3 es posible seleccionar uno de los complementos haciendo “click” sobre uno de ellos. Esto permite mostrar el nombre del complemento y su descripción en el panel de imágenes.</p>
Requerimientos	1. Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos
Comentario	<ol style="list-style-type: none"> 1. Para mostrar cada uno de los complementos, se muestra una imagen en miniatura del propio complemento y su correspondiente nombre. 2. Por defecto, si no se introduce ningún patrón de búsqueda, se muestran todos los complementos disponibles.

Tabla 18 – Caso de uso: Buscar complementos

6.16. Cargar complemento

Caso de Uso	CU16: Cargar complemento
Actor	Cliente, Administrador
Descripción	El usuario puede cargar uno de los complementos disponibles en el panel de edición para verlo con mayor detalle.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Complementos” dentro del panel de imágenes para mostrar la página de complementos.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	3. Seleccionar uno de los complementos visibles dentro de la página de complementos mediante un “doble click”.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de complementos En caso de estar visible la página de complementos el flujo comienza en FB.3.</p>
Pre-condiciones	<p>1. Entrada satisfactoria en la aplicación (CU2).</p> <p>2. Al menos un complemento visible en la página de complementos.</p>
Post-condiciones	1. El complemento se muestra en el panel de edición.
Requerimientos	Ninguno
Comentario	1. Los complementos visibles en la página de complementos dependerán de la búsqueda realizada. Ver CU15.

Tabla 19 – Caso de uso: Cargar complemento

6.17. Cargar imagen externa

Caso de Uso	CU17: Cargar complemento
Actor	Cliente, Administrador
Descripción	El usuario puede introducir dentro de la aplicación una imagen propia disponible en el ordenador para introducirla dentro de su composición.
Flujo básico (FB)	<p>1. Acceder al panel de composición mediante el menú o la barra de herramientas.</p> <p>2. Seleccionar la pestaña “Composición” dentro del panel de edición para mostrar la página de composición.</p> <p>3. Arrastrar un archivo de imagen externo a la aplicación en formato PNG sobre la parte superior del panel de edición.</p>
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de complementos En caso de estar visible la página de composición el flujo comienza en FB.3.</p>
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	1. La imagen externa se muestra en el panel de edición.
Requerimientos	1. Disponer de archivos de imagen en formato PNG en el ordenador.
Comentario	1. Las imágenes externas introducidas en la aplicación son procesadas aplicándoles un efecto de posterización que se asemeja al estilo de las composiciones propias de la aplicación.

Tabla 20 – Caso de uso: Cargar imagen externa



CAPÍTULO 3: DESARROLLO DEL PROYECTO

6.18. Añadir elemento a la composición

Caso de Uso	CU18: Añadir elemento a la composición
Actor	Cliente, Administrador
Descripción	El usuario puede añadir a su composición un elemento previamente cargado en el panel de edición.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Composición” dentro del panel de edición para mostrar la página de composición. 3. Presionar el botón “Añadir elemento”.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de composición En caso de estar visible la página de composición el flujo comienza en FB.3.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2). 2. Elemento cargado en el panel diseño. (CU14, CU16, CU17)
Post-condiciones	<ol style="list-style-type: none"> 1. El elemento se muestra integrado en la composición ubicada en el panel proyecto. 2. El elemento aparece en la página “Elementos” del panel de edición en forma de elemento modificable. 3. El elemento es eliminado de la página de “composición” del panel de edición.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. El elemento añadido aparece situado en la esquina superior izquierda de la composición, con una escala predeterminada y en el primer plano.

Tabla 21 – Caso de uso: Añadir elemento a la composición

6.19. Modificar posición

Caso de Uso	CU19: Modificar posición
Actor	Cliente, Administrador
Descripción	El usuario puede modificar la posición de uno de los elementos incluidos en la composición.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Elementos” dentro del panel de edición para mostrar la página de elementos. 3. Utilizar los <i>sliders</i> de posición en el eje X y posición en el eje Y (primero y segundo, respectivamente) para cambiar la posición que ocupa el elemento dentro de la composición.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza</p>



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	en FB.2. FA2. Vista de la página de elementos En caso de estar visible la página de elementos el flujo comienza en FB.3.
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2). 2. Al menos un elemento añadido a la composición (CU17).
Post-condiciones	1. El elemento correspondiente en la composición del panel proyecto altera su posición de acuerdo a los <i>sliders</i> de posición del panel de edición.
Requerimientos	Ninguno
Comentario	1. La imagen situada al lado de los <i>sliders</i> permite diferenciar a qué elemento de la composición corresponden.

Tabla 22 – Caso de uso: Modificar posición

6.20. Modificar tamaño

Caso de Uso	CU19: Modificar tamaño
Actor	Cliente, Administrador
Descripción	El usuario puede modificar el tamaño de uno de los elementos incluidos en la composición.
Flujo básico (FB)	1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Elementos” dentro del panel de edición para mostrar la página de elementos. 3. Utilizar el <i>slider</i> de tamaño (tercero) para cambiar el tamaño del elemento dentro de la composición.
Flujos alternativos (FA)	FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2. FA2. Vista de la página de elementos En caso de estar visible la página de elementos el flujo comienza en FB.3.
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2). 2. Al menos un elemento añadido a la composición (CU18).
Post-condiciones	1. El elemento correspondiente en la composición del panel proyecto altera su tamaño de acuerdo al <i>slider</i> de tamaño del panel de edición.
Requerimientos	Ninguno
Comentario	1. La imagen situada al lado del <i>slider</i> permite diferenciar a qué elemento de la composición corresponde. 2. Existen límites de tamaño mínimo y máximo

Tabla 23 – Caso de uso: Modificar tamaño



CAPÍTULO 3: DESARROLLO DEL PROYECTO

6.21. Cambiar de plano

Caso de Uso	CU21: Cambiar de plano
Actor	Cliente, Administrador
Descripción	El usuario puede cambiar de plano uno de los elementos incluidos en la composición.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Elementos” dentro del panel de edición para mostrar la página de elementos. 3. Utilizar los botones “Enviar atrás”, “Traer adelante”, “Enviar al fondo” y “Traer al frente” para cambiar de plano el elemento dentro de la composición.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de elementos En caso de estar visible la página de elementos el flujo comienza en FB.3.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2). 2. Al menos un elemento añadido a la composición (CU18). 3. Escenario cargado en la composición (CU12).
Post-condiciones	<ol style="list-style-type: none"> 1. El elemento correspondiente en la composición del panel proyecto cambia de plano.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. La imagen situada al lado de los botones permite diferenciar a qué elemento de la composición corresponden. 2. Botón “Enviar atrás”: Desplaza el elemento un plano hacia el fondo. 3. Botón “Traer adelante”: Desplaza el elemento un plano hacia el frente. 4. Botón “Enviar al fondo”: Desplaza el elemento hacia el fondo tantos planos como sea posible según el escenario cargado. 5. Botón “Traer al frente”: Desplaza el elemento hacia el frente hasta mostrarlo en primer plano.

Tabla 24 – Caso de uso: Cambiar de plano

6.22. Voltear horizontalmente

Caso de Uso	CU22: Voltear horizontalmente
Actor	Cliente, Administrador
Descripción	El usuario puede voltear horizontalmente (espejo) uno de los elementos incluidos en la composición.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Elementos” dentro del panel de edición



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	<p>para mostrar la página de elementos.</p> <p>3. Presionar el botón “Espejo” para voltear horizontalmente el elemento dentro de la composición.</p>
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de elementos En caso de estar visible la página de elementos el flujo comienza en FB.3.</p>
Pre-condiciones	<p>1. Entrada satisfactoria en la aplicación (CU2).</p> <p>2. Al menos un elemento añadido a la composición (CU18).</p>
Post-condiciones	<p>1. El elemento correspondiente en la composición del panel proyecto es volteado de forma horizontal.</p>
Requerimientos	<p>Ninguno</p>
Comentario	<p>1. La imagen situada al lado del botón “Espejo” permite diferenciar a qué elemento de la composición corresponde.</p>

Tabla 25 – Caso de uso: Voltear horizontalmente

6.23. Eliminar elemento de la composición

Caso de Uso	CU23: Eliminar elemento de la composición
Actor	Cliente, Administrador
Descripción	El usuario puede eliminar uno de los elementos de su composición.
Flujo básico (FB)	<p>1. Acceder al panel de composición mediante el menú o la barra de herramientas.</p> <p>2. Seleccionar la pestaña “Composición” dentro del panel de edición para mostrar la página de composición.</p> <p>3. Presionar el botón “Eliminar”.</p>
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de elementos En caso de estar visible la página de elementos el flujo comienza en FB.3.</p>
Pre-condiciones	<p>1. Entrada satisfactoria en la aplicación (CU2).</p> <p>2. Al menos un elemento añadido a la composición (CU18).</p>
Post-condiciones	<p>1. El elemento desaparece de la composición ubicada en el panel proyecto.</p> <p>2. El elemento se elimina de la página “Elementos” del panel de edición.</p>
Requerimientos	<p>Ninguno</p>
Comentario	<p>1. La modificación de las características del elemento ya no es posible.</p>

Tabla 26 – Caso de uso: Eliminar elemento de la composición



CAPÍTULO 3: DESARROLLO DEL PROYECTO

6.24. Editar un efecto

Caso de Uso	CU24: Editar un efecto
Actor	Cliente, Administrador
Descripción	El usuario puede ajustar los valores de algunas propiedades fotográficas en la composición que está diseñando.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Composición” dentro del panel de edición para mostrar la página de composición. 3. Utilizar los <i>sliders</i> para modificar los valores de las propiedades fotográficas y la propia composición.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de composición En caso de estar visible la página de composición el flujo comienza en FB.3.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2). 2. Escenario cargado en la composición (CU12).
Post-condiciones	1. La composición del panel proyecto se ve afectada por las propiedades fotográficas modificadas.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. Se consideran como propiedades fotográficas para este CU el “Brillo”, “Contraste”, “Color” y “Nitidez”. 2. Las propiedades pueden variarse entre los valores -100 y 100, y por defecto son 0.

Tabla 27 – Caso de uso: Editar un efecto

6.25. Aplicar un filtro

Caso de Uso	CU25: Aplicar un filtro
Actor	Cliente, Administrador
Descripción	El usuario puede aplicar filtros fotográficos a la composición que está diseñando.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Acceder al panel de composición mediante el menú o la barra de herramientas. 2. Seleccionar la pestaña “Composición” dentro del panel de edición para mostrar la página de composición. 3. Utilizar los distintos botones de filtro situados en el panel “Efectos y filtros” para activarlos.
Flujos alternativos (FA)	<p>FA1. Vista del panel de composición En caso de estar visible el panel de composición el flujo comienza en FB.2.</p> <p>FA2. Vista de la página de composición En caso de estar visible la página de composición el flujo comienza</p>



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	<p>en FB.3.</p> <p>FA2. Filtro activo</p> <p>En caso de ya estar activos algunos de los filtros el flujo continúa a partir de FB.2 pudiendo presionar los botones de filtro para desactivarlos.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2). 2. Escenario cargado en la composición (CU12).
Post-condiciones	<ol style="list-style-type: none"> 1. La composición del panel proyecto se ve afectada por los filtros fotográficos activos.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. Se considera como filtro la acción “voltear horizontalmente” en la explicación de este CU aunque no sea realmente un filtro de fotografía. 2. El resto de filtros fotográficos son: “posterizar”, “equalizar” y “alisar”.

Tabla 28 – Caso de uso: Aplicar un filtro

6.26. Crear nuevo proyecto

Caso de Uso	CU26: Crear nuevo proyecto
Actor	Cliente, Administrador
Descripción	El usuario puede crear un nuevo proyecto vacío, eliminando la composición actual del proyecto.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Presionar el botón de “Nuevo proyecto” en el menú o la barra de herramientas. 2. Presionar “Si” en la ventana de nuevo proyecto para aceptar la pérdida de cambios no guardados y la apertura de un nuevo proyecto en blanco.
Flujos alternativos (FA)	<p>FA1. Cancelar en la ventana de nuevo proyecto</p> <p>En FB.2 en lugar de presionar el botón “Si” el usuario presiona el botón “No” a causa de no aceptar las condiciones. El flujo termina inmediatamente sin cumplirse las post-condiciones.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	<ol style="list-style-type: none"> 1. La composición creada en el panel proyecto se elimina y vuelve a aparecer el lienzo de proyecto vacío. 2. Los <i>sliders</i> de efecto situados en la página “Composición” del panel de edición vuelven a su posición por defecto. 3. Los botones de filtro situados en la página “Composición” del panel de edición pasan a estado “desactivado”. 4. Todos elementos situados en la página “Elementos” del panel de edición se eliminan.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. Los cambios realizados en cualquier proyecto no guardado no podrán volver a recuperarse.

Tabla 29 – Caso de uso: Crear nuevo proyecto



CAPÍTULO 3: DESARROLLO DEL PROYECTO

6.27. Guardar proyecto

Caso de Uso	CU27: Guardar proyecto
Actor	Cliente, Administrador
Descripción	El usuario puede guardar el proyecto activo en la aplicación para posteriormente recuperarlo y seguir editando en él.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Presionar el botón de “Guardar proyecto” en el menú o la barra de herramientas. 2. Seleccionar la carpeta donde será guardado el proyecto en la ventana de guardar proyecto. 3. Escribir un nombre para el proyecto guardado en la ventana de guardar proyecto. 4. Presionar “Guardar” en la ventana de guardar proyecto. 5. Presionar “Ok” en la nueva ventana de guardar proyecto para confirmar los cambios.
Flujos alternativos (FA)	<p>FA1. Cancelar o cerrar la ventana de guardar proyecto En FB.2 o FB.3 se presiona el botón “Cancelar” o se cierra la ventana de guardar proyecto. El flujo termina inmediatamente sin cumplirse las post-condiciones.</p> <p>FA2. Sobrescribir proyecto guardado En FB.3 se selecciona uno de los archivos del directorio que ya contiene un proyecto guardado o se escribe el mismo nombre que uno de los archivos existente en ese directorio. Pulsar el botón “Guardar” en la ventana de guardar proyecto. Presionar el botón “Si” para aceptar el reemplazamiento del archivo de guardado y “Ok” en la nueva ventana de guardar proyecto para confirmar los cambios.</p>
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	1. Se crea o reemplaza un archivo “.ad” en el directorio seleccionado almacenando la configuración del proyecto que se estaba editando en ese momento (escenario, elementos y sus propiedades, efectos, filtros,...).
Requerimientos	Ninguno
Comentario	1. Además de almacenar los datos relativos a la composición en el archivo de guardado, se guarda también el usuario que ha llevado a cabo la edición de la composición.

Tabla 30 – Caso de uso: Guardar proyecto

6.28. Abrir Proyecto

Caso de Uso	CU28: Abrir proyecto
Actor	Cliente, Administrador
Descripción	El usuario puede abrir un proyecto previamente guardado para continuar con la edición de su composición.
Flujo básico (FB)	1. Presionar el botón de “Abrir proyecto” en el menú o la barra de herramientas.



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	<ol style="list-style-type: none"> 2. Confirmar la apertura de un nuevo proyecto perdiendo los cambios no guardados en el proyecto actual. 3. Seleccionar un directorio donde existan proyectos guardados. 4. Escribir el nombre de uno de los proyectos guardados en el directorio o seleccionar directamente uno de ellos. 5. Presionar “Abrir” en la ventana de guardar proyecto.
Flujos alternativos (FA)	<p>FA1. Cancelar o cerrar la ventana de abrir proyecto En FB.3 o FB.4 se presiona el botón “Cancelar” o se cierra la ventana de abrir proyecto. El flujo termina inmediatamente sin cumplirse las post-condiciones.</p> <p>FA2. Escribir el nombre de un proyecto no existente En FB.4 el usuario escribe el nombre de un proyecto que no existe en el directorio seleccionado en FB.3. Después presiona el botón “Abrir”. El flujo termina inmediatamente sin cumplirse las post-condiciones.</p> <p>FA3. Usuario incorrecto En FB.4 se selecciona para abrir un proyecto guardado por otro usuario. Se presiona el botón “Abrir”. Una ventana notifica la situación al usuario. El flujo termina inmediatamente sin cumplirse las post-condiciones.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2). 2. Proyecto guardado previamente por el usuario activo en la aplicación (CU27).
Post-condiciones	<ol style="list-style-type: none"> 1. La composición guardada en el archivo seleccionado se carga en la composición del panel proyecto. 2. Los valores de los efectos guardados en el archivo seleccionado se cargan en los <i>sliders</i> de efecto situados en la página “Composición” del panel de edición. 3. Se actualizan los botones correspondientes a los filtros situados en la página “Composición” del panel de edición a estado “activado” o “desactivado” según los datos de proyecto abierto. 4. Todos elementos guardados en el proyecto abierto se muestran en la página “Elementos” del panel de edición con sus propiedades actualizadas según el proyecto abierto.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. El usuario puede continuar editando su composición al igual que lo haría antes de realizar el guardado del proyecto.

Tabla 31 – Caso de uso: Abrir proyecto

6.29. Exportar composición

Caso de Uso	CU29: Exportar composición
Actor	Cliente, Administrador
Descripción	El usuario puede exportar de la aplicación la composición creada en uno de los formatos de imagen propuestos.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Presionar el botón de “Exportar composición” en el menú. 2. Seleccionar la carpeta donde será guardada la imagen con la



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	<p>composición en la ventana de exportar composición.</p> <p>3. Escribir un nombre para el archivo de imagen guardado en la ventana de exportar composición.</p> <p>4. Presionar “Guardar” en la misma ventana..</p> <p>5. Pulsar “Ok” en la nueva ventana para confirmar los cambios.</p>
Flujos alternativos (FA)	<p>FA1. Cancelar o cerrar la ventana de exportar composición En FB.2 o FB.3 se presiona el botón “Cancelar” o se cierra la ventana de exportar composición. El flujo termina inmediatamente sin cumplirse las post-condiciones.</p> <p>FA2. Sobrescribir archivo En FB.3 se selecciona uno de los archivos de imagen del directorio o se escribe el mismo nombre que uno de los archivos existente en ese directorio. Pulsar el botón “Guardar” en la ventana de guardar proyecto. Presionar el botón “Si” para aceptar el reemplazamiento del archivo de imagen y “Ok” en la nueva ventana de exportar composición para confirmar los cambios.</p>
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	<p>1. Se crea o reemplaza un archivo de imagen en el directorio seleccionado con una imagen de la composición diseñada en ese momento.</p> <p>2. Un programa externo (el asignado por defecto para cada tipo de archivo) abre el archivo de imagen directamente para mostrar al usuario el resultado final de su composición.</p>
Requerimientos	Disponer de un programa instalado capaz de abrir archivos de imagen.
Comentario	1. Los formatos de imagen disponibles para exportar la composición son PNG y JPEG.

Tabla 32 – Caso de uso: Exportar composición

6.30. Abrir proyecto de usuario

Caso de Uso	CU29: Abrir proyecto de usuario
Actor	Cliente, Administrador
Descripción	El usuario puede abrir uno de sus proyectos previamente guardados desde el panel de usuario para continuar con la edición de su composición.
Flujo básico (FB)	<p>1. Acceder al panel de usuario mediante el menú o la barra de herramientas.</p> <p>2. Seleccionar mediante un “doble click” la imagen de la composición de uno de sus proyectos guardados para seleccionar el archivo que contiene proyecto guardado correspondiente.</p>
Flujos alternativos (FA)	<p>FA1. Vista del panel de usuario En caso de estar visible el panel de usuario el flujo comienza en FB.2.</p>
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).



CAPÍTULO 3: DESARROLLO DEL PROYECTO

	2. Proyecto guardado previamente por el usuario activo en la aplicación (CU27).
Post-condiciones	<ol style="list-style-type: none"> 1. El panel de usuario se reemplaza por el panel de composición. 2. La composición guardada en el archivo seleccionado se carga en la composición del panel proyecto. 3. Los valores de los efectos guardados en el archivo seleccionado se cargan en los <i>sliders</i> de efecto situados en la página “Composición” del panel de edición. 4. Se actualizan los botones correspondientes a los filtros situados en la página “Composición” del panel de edición a estado “activado” o “desactivado” según los datos de proyecto abierto. 5. Todos elementos guardados en el proyecto abierto se muestran en la página “Elementos” del panel de edición con sus propiedades actualizadas según el proyecto abierto.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. El usuario puede continuar editando su composición al igual que lo haría antes de realizar el guardado del proyecto. 2. Los proyectos mostrados en el panel de usuario son solo aquellos ubicados en directorio “/data/saves” de la aplicación y que han sido creados por el usuario activo en la aplicación. 3. Para mostrar cada uno de los proyectos, se muestra una imagen en miniatura de la composición del proyecto guardado y el correspondiente nombre del archivo que contiene el proyecto.

Tabla 33 – Caso de uso: Abrir proyecto de usuario

6.31. Cambiar de usuario

Caso de Uso	CU31: Cambiar de usuario
Actor	Cliente, Administrador
Descripción	El usuario puede salir parcialmente de la aplicación para entrar como otro usuario diferente sin que ello suponga parar la ejecución del programa.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Presionar el botón de “Cambiar de usuario” en el menú o la barra de herramientas. 2. Rellenar los campos de usuario y clave de la ventana de inicio. 3. Presionar el botón “Entrar”.
Flujos alternativos (FA)	<p>FA1. Datos de entrada incorrectos En FB.3 la aplicación detecta que el usuario ha introducido alguno de los campos del formulario de forma incorrecta. Se vuelve a FB.2. El usuario debe volver a introducir los datos de forma correcta.</p> <p>FA2. Datos de entrada recordados En caso de que se haya activado la opción “Guardar el usuario al cerrar el programa” en las preferencias (CU7), si el usuario lo desea puede omitir el paso FB.2.</p>
Pre-condiciones	1. Entrada satisfactoria en la aplicación (CU2).



CAPÍTULO 3: DESARROLLO DEL PROYECTO

Post-condiciones	<ol style="list-style-type: none"> 1. La ventana de inicio se cierra. 2. La ventana de bienvenida de abre (<i>splashscreen</i> en forma de vídeo) durante el período de carga. 3. Tras la carga completa de la aplicación la ventana principal se abre con la sesión del correspondiente al nuevo usuario introducido. Si se ha activado anteriormente la opción “Guardar la vista al cerrar el programa” en las preferencias (CU7), la ventana principal se cargará con el aspecto que tenía en el momento de cierre de la última ejecución. 4. En caso de que se haya activado la opción “Abrir último proyecto guardado al iniciar el programa” en las preferencias (CU7), se cargará el último proyecto guardado si pertenece al usuario que ha iniciado sesión.
Requerimientos	<ol style="list-style-type: none"> 1. Es necesaria conexión a Internet para comunicarse con el servidor que contiene la base de datos.
Comentario	<ol style="list-style-type: none"> 1. La aplicación coteja los datos de usuario y clave con la base de datos. Si alguno de los datos introducidos no se corresponde se muestra una nota indicando al usuario el problema, el campo se vuelve de color rojo, y el cursor de escritura se sitúa sobre dicho campo.

Tabla 34 – Caso de uso: Cambiar de usuario

6.32. Salir de la aplicación

Caso de Uso	CU32: Cambiar de usuario
Actor	Cliente, Administrador
Descripción	El usuario salir de la aplicación en cualquier momento durante la ejecución del mismo.
Flujo básico (FB)	<ol style="list-style-type: none"> 1. Presionar el botón de “Cerrar” en el menú o cerrar una de las ventanas principales. 2. Confirmar el cierre del programa sin guardar pulsando el botón “Si” de la ventana “Cerrar”.
Flujos alternativos (FA)	<p>FA1. Cancelar cierre</p> <p>En FB.2 en caso de no confirmar el cierre presionando el botón “No” o cerrando la ventana de cierre el flujo termina inmediatamente sin cumplirse las post-condiciones.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. Entrada satisfactoria en la aplicación (CU2).
Post-condiciones	<ol style="list-style-type: none"> 1. Cualquier funcionalidad que esté llevándose a cabo finaliza de forma inmediata sin éxito. 2. La aplicación se cierra completamente (se cierran todas las ventanas) y su proceso finaliza.
Requerimientos	Ninguno
Comentario	<ol style="list-style-type: none"> 1. Salir de la aplicación implica para el usuario perder los cambios realizados en el proyecto actual que no hayan sido guardados.

Tabla 35 – Caso de uso: Salir de la aplicación



CAPÍTULO 3: DESARROLLO DEL PROYECTO

Algunas de las funcionalidades mencionadas disponen de accesos directos de teclado. Estos accesos están indicados en las distintas opciones que disponen el menú y los submenús.

7. CONEXIÓN APLICACIÓN – SERVIDOR

Para que la aplicación pueda llevar a cabo las funcionalidades descritas debe ser capaz de relacionarse con la base de datos, para ello es necesaria una conexión de la aplicación con el servidor que mantiene la base de datos.

Para llevar a cabo esta conexión se he hecho uso de la arquitectura software de transferencia REST, el servidor MySQL de XAMPP, conexiones TCP/IP, el lenguaje de bases de datos SQL y la librería de Python MySQLdb. La comunicación se lleva a cabo mediante tres pasos:

Conexión: En el fichero “db.py” se describe la conexión y los parámetros adecuados para acceder a la base de datos haciendo uso de las funciones que ofrece la librería MySQLdb. Los parámetros necesarios en nuestro caso son:

host: Permite indicar la dirección donde se encuentra el servidor de la base de datos. Se puede indicar el nombre del host o la dirección IP donde se encuentra.

port: Es el puerto de la conexión TCP/IP para el servidor MySQL. Por defecto, MySQL escucha las peticiones que llegan a través del puerto 3306.

user: Se indica el nombre de usuario que pretende autenticarse en el servidor MySQL para acceder a los servicios que este ofrece. Dependiendo del usuario puede tener acceso a unas u otras bases de datos y tener determinados permisos sobre ellas.

password: Es la contraseña o clave que permite la correcta autenticación del usuario en el servidor.

db: Se utiliza para indicar el nombre de la base de datos concreta a la que queremos acceder dentro de las posibles en el servidor MySQL.

Una vez ha sido establecida correctamente la conexión, se crea un *socket* TCP/IP para la transferencia de información.

CAPÍTULO 3: DESARROLLO DEL PROYECTO

Transferencia: En esta fase se lleva a cabo la comunicación con la base de datos. Una vez está abierto el *socket* mediante la fase de conexión, se pueden realizar conexiones a modo de consultas o peticiones mediante el lenguaje SQL. De esta forma la aplicación se está comunicando directamente con la base de datos, pudiendo recibir información de ella o incluso modificar sus datos o estructura.

Socket: Es un método para la comunicación entre un programa del cliente y un programa del servidor en una red. Un socket se define como el punto final en una conexión. Los sockets se crean y se utilizan con un sistema de peticiones o de llamadas de función a veces llamados interfaz de programación de aplicación de *sockets* (*API, application programming interface*). Permiten intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

Cierre: Tras la fase de de transferencia el último paso es cerrar la conexión. Mediante otra función de la librería MySQLdb se hace de forma sencilla. Esta finaliza la conexión y el envío de datos.

Durante el desarrollo de la aplicación se ha trabajado con una versión local de la aplicación, es decir, ejecutando las versiones de prueba de la aplicación en la misma máquina en la que está siendo ejecutado el servidor MySQL. Para este caso los parámetros relativos a la conexión el servidor son:

host: localhost / 127.0.0.1
port: 3306

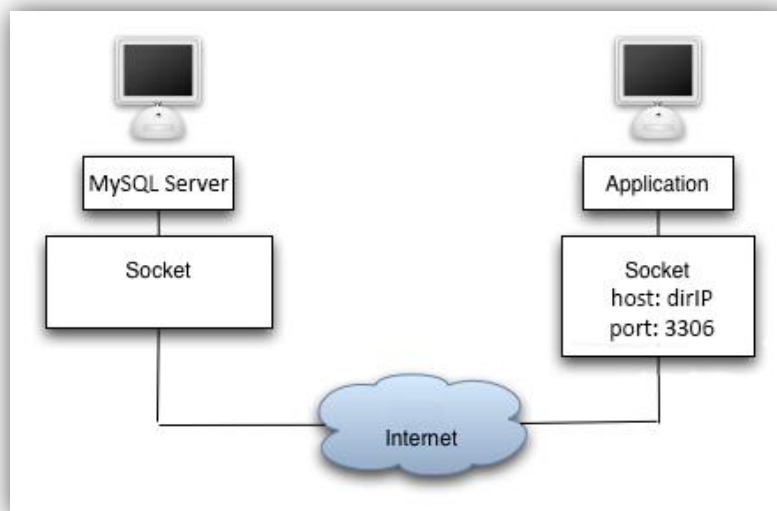


Ilustración 29– Conexión aplicación - servidor MySQL

El primero hace referencia a la máquina actual en la que se está ejecutando el programa y el servidor simultáneamente. Se puede indicar mediante su nombre de host (*localhost*) o mediante la dirección IP de la propia máquina: 127.0.0.1. En el segundo, se indica el puerto



CAPÍTULO 3: DESARROLLO DEL PROYECTO

3306, el predeterminado para el servidor MySQL. Se puede ver esta configuración en la *Ilustración 22*.

Sin embargo, para la versión final de la aplicación, ésta debe ser capaz de conectarse al servidor desde cualquier máquina con acceso a Internet. Es decir, necesitamos una conexión remota con el servidor MySQL. La configuración para este caso puede volverse más compleja. Los parámetros de conexión con el servidor quedarían de la siguiente forma:

host: [dirección IP]
port: 3306 u otro

En el parámetro *host* debemos indicar la dirección IP de la máquina donde se encuentra el servidor o la dirección del *router* que gestiona la red donde se encuentra el servidor. Y en el parámetro *port* el puerto predeterminado 3306 en el caso de indicar directamente la dirección de la máquina del servidor, u otro puerto según la configuración de la red en el caso de indicar la dirección del *router*. La implementación de esta configuración se explica con mayor detalle en el anexo ubicado al final del documento.

Para los parámetros relacionados con el propio servidor y la base de datos debemos indicar primero el usuario (*user*) y clave (*password*) establecidos para acceder al servidor MySQL. Mediante la herramienta phpMyAdmin es fácil crear y modificar nuevos usuarios para las bases de datos. Es necesario también establecer el parámetro *-db* como 'autodesign', puesto que es el nombre que recibe la base de datos de la aplicación.

8. CARACTERÍSTICAS

8.1. Responsividad

El término *responsividad* se refiere a la capacidad de una aplicación de reajustar su tamaño y apariencia según la pantalla del dispositivo o máquina. Y no solo eso, el usuario debe tener la posibilidad de redimensionar la aplicación independientemente de la resolución y tamaño de la pantalla.

En una aplicación de escritorio, como es el caso de la aplicación Autodesign, es frecuente que los usuarios decidan dimensionar y posicionar las ventanas de la aplicación en la pantalla a su gusto.

Centrándonos en la aplicación Autodesign, su ventana principal de la interfaz gráfica de usuario es redimensionable y de posición libre. Es decir, el usuario es capaz de fijar la posición y tamaño de la ventana en su pantalla ajustándola a sus necesidades o gustos. La aplicación tiene un tamaño mínimo de 800x550 píxeles y por defecto se muestra en el centro de la pantalla y en el tamaño descrito.

CAPÍTULO 3: DESARROLLO DEL PROYECTO

Modificar el tamaño de la interfaz implica, a su vez, que la apariencia de los elementos contenidos dentro de la ventana principal también se redimensionen o cambien su posición. Con esto se consigue mantener siempre una buena apariencia de la aplicación y mantener la usabilidad, puesto que se mantienen visibles todos los contenidos. Puesto que se trata de una aplicación de tratamiento de imágenes, su interfaz hace uso de muchos elementos formados por imágenes, y es capaz de ampliarlas, reducir las y/o colocarlas con precisión de acuerdo al espacio disponible para ellas dentro de las distintas ventanas o paneles.

En el apartado 4.3 de este capítulo, dedicado a la ventana principal de la interfaz, se explica la división de esta ventana en diferentes secciones, las cuales contienen otros paneles de la aplicación. La ventana principal tiene la capacidad de dimensionar estos paneles internos que contiene. De esta forma el usuario puede configurar la apariencia interna de la propia ventana, según la sección de la aplicación en la que esté trabajando o según sus gustos. Para ello solo debe seleccionar el separador situado entre distintos paneles y arrastrarlo a otra posición. Esta ventaja ofrece al usuario una mayor libertad a la hora de trabajar, mejorando la usabilidad de la aplicación, puesto que permite realizar algunas acciones con mayor facilidad o precisión. En la *Ilustración 30*, que se muestra a continuación, en la configuración de la izquierda se muestra el panel de edición con un mayor tamaño lo que permite apreciar mejor el elemento que va a insertarse en la composición o asignar valores más precisos mediante los sliders de efectos de la composición. Mientras que en la imagen situada a la derecha, el panel de imágenes ha sido ampliado para poder mostrar un mayor número de personajes en pantalla mientras se realiza una búsqueda.



Ilustración 30- Responsividad de la aplicación

En la funcionalidad de preferencias que ofrece la aplicación se muestra una opción relacionada con el dimensionamiento de la aplicación. Ésta nos permite activar el guardado de la posición, tamaño y estructura interna de paneles de la ventana principal



CAPÍTULO 3: DESARROLLO DEL PROYECTO

para la próxima ejecución del programa. Con ella activada, cuando la aplicación arranca lo hace con la misma apariencia que tenía en el momento de cierre de la última ejecución.

Todas las ventanas de a aplicación permiten posicionarlas con total libertad, pero no todas son redimensionables. Las ventanas dedicadas a las funciones de guardar, cargar y exportar composición sí permiten cambios de tamaño. Mientras que las ventanas de inicio, registro, email, preferencias y las distintas ventanas de control son de tamaño fijo, aunque su escalado tampoco ofrecería ninguna ventaja añadida en estos casos.

Añadir estas características responsivas a la aplicación ha sido una de las tareas más difíciles de llevar a cabo. Por una parte debido a la complejidad que ello requiere, por otra parte por arrancar desde cero en el lenguaje de programación Python con una librería gráfica (wxPython) bastante compleja y, finalmente, por mi escasa experiencia en el diseño de interfaces gráficas.

8.2. Usabilidad

La usabilidad, en el ámbito del desarrollo software, se refiere a la facilidad con la que los usuarios de una aplicación o herramienta pueden utilizarla. Claridad, adaptabilidad, facilidad o efectividad son algunos de los términos que vienen ligados a este concepto.

La aplicación Autodesign ha sido diseñada de forma orientada al usuario final de la aplicación. Se pretende conseguir la aplicación tenga un manejo sencillo para la mayor parte de usuarios, o que al menos tenga un fácil aprendizaje. Se tiene como objetivo también que la interfaz de la aplicación sea clara, organizada y estéticamente agradable, favoreciendo así la usabilidad. La adaptabilidad también ha sido un punto clave para intentar mejorar la facilidad de uso de la aplicación.

Algunos de los elementos que han conseguido mejorar la usabilidad de la aplicación se citan a continuación:

- i. Las notas informativas que aparecen en ventanas como la de inicio o registro permiten informar al usuario de los problemas ocurridos en la introducción de datos en formularios. Se especifica de forma clara cuál es el campo en el que los datos correctos y se indica la forma de introducirlos correctamente.
- ii. Siguiendo con el punto anterior, cuando un campo no es correcto aparece remarcado en color rojo y el cursor de escritura se posiciona directamente sobre él para que pueda modificarlo directamente.
- iii. El video que aparece como pantalla de bienvenida a la aplicación (*splashscreen*) informa al usuario de que la aplicación está en proceso de



CAPÍTULO 3: DESARROLLO DEL PROYECTO

carga y pretende amenizarlo durante la espera. La aplicación dispone incluso de dos videos que aparecen alternadamente de forma aleatoria.

- iv. La barra de herramientas constituye en sí misma un elemento que favorece la usabilidad, poniendo al alcance del usuario algunas de las funciones y manteniéndolas siempre visibles.
- v. Los mensajes de control; tanto informativos, de error o de preguntas; permiten al usuario tomar decisiones sobre el programa o informarse de algunos sucesos.
- vi. Las barras de carga mejoran también la experiencia del usuario, puesto que sin ellas un usuario no sabe si se está realizando una determinada tarea o cual es el tiempo estimado para su ejecución. En algunas circunstancias, si no aparece una barra de progreso o elemento similar, el usuario puede pensarse que la aplicación ha quedado bloqueada.
- vii. Las preferencias de usuario permiten al usuario alterar algunas de las funcionalidades o la estética de la aplicación a su gusto. Esta configuración personal permite mejorar la satisfacción de los usuarios.
- viii. La libertad de dimensionamiento de la interfaz de la aplicación por parte de los usuarios explicada anteriormente constituye un factor muy importante de cara a la usabilidad. Permite configurar la estética de la aplicación y llevar a cabo tareas de una forma más agradable, rápida y efectiva.
- ix. Algunas funcionalidades realizan un cambio de panel automático. Por ejemplo, cuando un usuario carga un proyecto desde el panel de usuario, automáticamente se muestra el panel de composición para que el resultado sea visible de forma inmediata y pueda seguir trabajando directamente.
- x. Accesos directos de teclado para las funcionalidades que aparecen en el menú y los submenús.

Estos son algunos de las características de la aplicación que permiten ofrecer una mejor experiencia al usuario, mejorando así la calidad final del programa.

9. PAQUETE DE INSTALACIÓN

Como ya se ha mencionado anteriormente la aplicación ha sido diseñada con un enfoque hacia los usuarios finales de la aplicación. También se ha comentado el concepto de “usabilidad” dentro de la aplicación, y es por estas razones por las que se ha creado un paquete de instalación. La aplicación consta de un paquete de instalación que va a permitir organizar y configurar los archivos necesarios para la ejecución de la aplicación dentro de la máquina del cliente o usuario de la una forma rápida y sencilla. El usuario final no tiene por qué conocer las



CAPÍTULO 3: DESARROLLO DEL PROYECTO

configuraciones o archivos necesarios para que la aplicación pueda funcionar con normalidad, y es por ello que debemos poner a su disponibilidad una versión de la aplicación que no requiera de estos conocimientos.

Para lograr este objetivo se han creado diferentes archivos y los pasos necesarios se recogen en los apartados mostrados a continuación.

9.1. Archivo ejecutable

Para que un usuario pueda ejecutar la aplicación de la forma habitual es necesario crear un archivo ejecutable. Este archivo tendrá la extensión “.exe” ya que el uso de esta aplicación está restringido a equipos que utilicen un sistema operativo Windows. Este archivo podrá, además, ser ejecutado en equipos que no tengan instalado el lenguaje de programación Python.

Para crear este archivo se utiliza una librería de Python llamada *py2exe*. Esta librería externa proporciona los métodos y módulos necesarios para configurar y crear un ejecutable a partir de uno o varios archivos de Python. Para ello es necesario crear un script independiente, el cual se ha llamado “setup.py”.

En este se especifican algunas características de la aplicación como el nombre de la misma o el autor. También se pueden indicar algunas propiedades de los archivos de salida como su compresión, nombre, carpeta de destino, descripción o incluso icono gráfico. Se deben seleccionar los archivos Python que se desean convertir y se pueden cambiar algunas opciones más avanzadas mediante la librería *py2exe*. Es necesario mencionar, que este archivo no genera exclusivamente un archivo “.exe”, sino que genera además una serie de archivos de los que depende.

La ejecución del archivo “setup.py” mediante la línea de comandos genera como salida dos carpetas o directorios. La carpeta “build”, la cual contiene archivos temporales necesarios para llevar a cabo este proceso y por lo cual puede ser eliminada una vez terminado. Y una carpeta con nombre predeterminado “dist” (en mi caso ha sido cambiado a “Autodesign 1.0”) que contiene tanto el archivo ejecutable como los archivos de dependencias. Estos últimos son ficheros “.pyd” (archivos dinámicos de Python) y “.dll” (archivos de bibliotecas que contienen código), y son estrictamente necesarios para que el archivo “.exe” pueda funcionar.

9.2. Instalador

Contar con un instalador facilita a los usuarios finales organizar los archivos necesarios para que la aplicación funcione correctamente, ya que es el propio instalador el que lo hace.

CAPÍTULO 3: DESARROLLO DEL PROYECTO

Para crear el paquete de instalación se ha hecho uso de la herramienta de compresión de archivos *WinRAR*. Utilizando esta herramienta podemos generar un archivo autoextraíble que funcionará como el instalador de la aplicación. El programa no funciona exclusivamente con los archivos generados mediante el proceso del apartado anterior, sino que también necesita de otros recursos. Añadiremos al paquete además algunos de los directorios y ficheros mencionados anteriormente en el apartado 2 de este mismo capítulo. El paquete de instalación contendrá:

- i. La carpeta “Autodesign 1.0” generada en el apartado anterior, con el ejecutable y los demás archivos de los que depende.
- ii. La carpeta “img” que contiene todas las imágenes.
- iii. La carpeta “video” que contiene los vídeos de la aplicación.
- iv. La carpeta “data” donde se guardan los archivos y composiciones guardadas.
- v. La carpeta “config” que contiene los archivos de configuración.
- vi. El manual de usuario.

El paquete de instalación tendrá también forma de ejecutable “.exe”. Cuando sea ejecutado aparecerá un programa de instalación como el de la *Ilustración 31*, el cual descomprime los archivos, los extrae y los copia en una carpeta específica de la máquina del usuario (por defecto en C:\Archivos de programa). Permite crear además un acceso directo al programa, es decir, al archivo “Autodesign.exe” que lanza la aplicación.

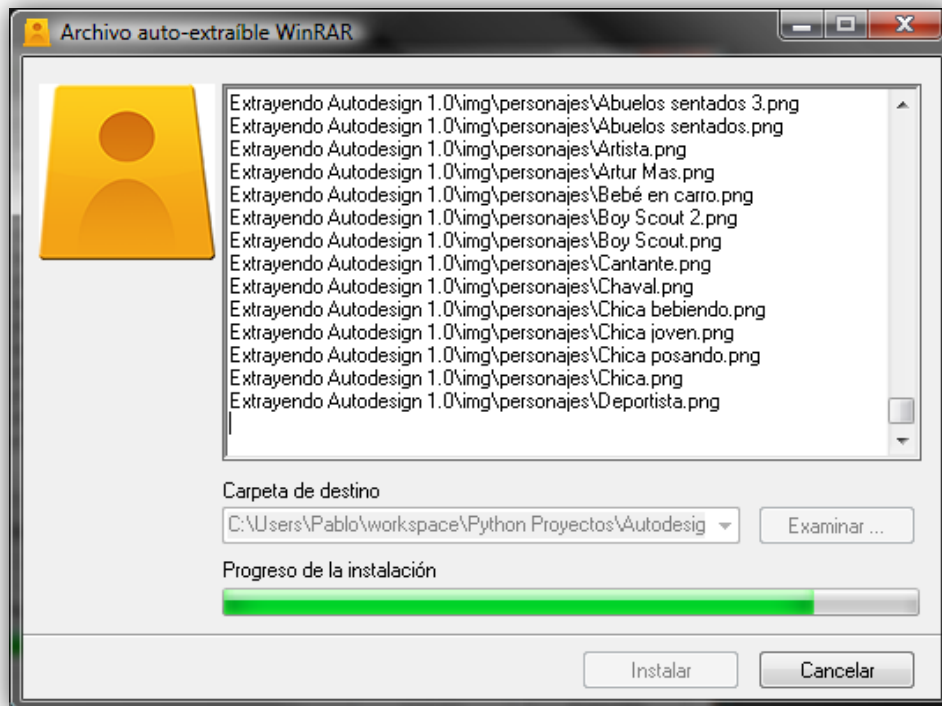


Ilustración 31 – Instalador



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio
dedicada a la composición fotográfica

CAPÍTULO 3: DESARROLLO DEL PROYECTO



CAPÍTULO 4: MANUAL DE USUARIO

En el presente capítulo se recogen una serie de indicaciones para el uso y correcto manejo de la aplicación. Se considera la aplicación Autodesign como una aplicación bastante intuitiva, aún así se incorpora el presente manual de usuario para que toda la información sobre su uso esté disponible para cualquier usuario que lo pudiese necesitar. Además, la aplicación contiene tanto en el menú como en la barra de herramientas botones que permite al usuario acceder directamente al manual de instrucciones.

Para solicitar ayuda adicional póngase en contacto con el administrador de la aplicación a través de correo electrónico mediante la opción “Enviar correo” de la propia aplicación.

1. GUÍA DE INSTALACIÓN

Para proceder con la instalación de la aplicación Autodesign realice los siguientes pasos:

1. Ejecute el archivo de instalación “Autodesign 1.0 – Installer” en su ordenador.
2. Seleccione una carpeta de destino (será el directorio donde se almacene la aplicación). Puede realizarlo de dos formas:
 - a. Presione el botón “Examinar” y navegue por las carpetas de forma gráfica. Seleccione una carpeta y pulse “Aceptar”.
 - b. Introduzca manualmente mediante el teclado una ruta de destino.

Por defecto la carpeta de destino es: “C:\Program Files\Autodesign” [Nota]

Si la carpeta de destino no existe será creada automáticamente antes de la instalación.

3. Pulse el botón “Instalar”.
4. Espere unos minutos durante el proceso de instalación.
5. El programa de instalación se cerrará automáticamente cuando el proceso finalice.



CAPÍTULO 4: MANUAL DE USUARIO

6. Han sido creados accesos directos a la aplicación Autodesign en su escritorio y en el menú Inicio/programas.
7. Puede comenzar a utilizar Autodesign.

[Nota]: La carpeta “C:\Program Files\” y sus subcarpetas suelen necesitar permisos especiales de escritura. Si desea instalar aquí el programa es posible que durante el proceso de instalación se soliciten permisos de “Administrador”, y debe aceptar para continuar con la instalación. Además, el programa debe ser ejecutado como “Administrador” para que funcione correctamente en este directorio. Para ello haga “click” con el botón derecho del ratón sobre el archivo “Autodesign.exe” situado en la carpeta de instalación y seleccione la opción “Ejecutar como administrador”.

2. USUARIO

2.1. Registro

Registrarse supone darse de alta en los servicios que ofrece la aplicación. Para registrarse en la aplicación:

1. Pulse el botón “Registrarse” en la ventana de inicio.
2. Rellene todos los campos solicitados en el formulario de registro.
3. Si alguno de los campos no se ha introducido correctamente siga las instrucciones que aparecen sobre la propia ventana.
4. Pulse el botón “Registrarse” de la propia ventana de registro.
5. Si el proceso ha finalizado de forma correcta aparecerá una ventana de confirmación.


2.2. Entrar

1. Rellene los campos de usuario y clave de la ventana de inicio con los datos de correspondientes de su registro.
2. Si alguno de los campos no se ha introducido correctamente siga las instrucciones que aparecen sobre la propia ventana.
3. Pulse el botón “Entrar” de la propia ventana de inicio.
4. Si el proceso ha finalizado de forma correcta aparecerá la ventana principal de la aplicación.



CAPÍTULO 4: MANUAL DE USUARIO

2.3. Modificar tus datos

1. Acceda al panel de usuario mediante el submenú “Herramientas” situado en el menú superior, seleccione “Mostrar/Ocultar” y después “Usuario”. También puede acceder mediante el botón  situado la barra de herramientas o la combinación de teclas “Ctrl+U”.
2. Aparecerá un formulario con sus datos correspondientes en la parte izquierda del panel de usuario. Modifique todos los campos que desee sobre el propio formulario.
3. Si alguno de los campos no se ha introducido correctamente siga las instrucciones que aparecen sobre la propia ventana.
4. Pulse el botón “Actualizar los datos” de la propia ventana de inicio.
5. Si el proceso ha finalizado de forma correcta aparecerá una ventana de confirmación.

Para cambiar su clave de usuario:

1. Presione sobre el botón “Cambiar clave” situado en el panel de usuario.
2. En la nueva ventana introduzca su clave anterior y su clave nueva en el formulario.
3. Si alguno de los campos no se ha introducido correctamente siga las instrucciones que aparecen sobre la propia ventana.
4. Pulse el botón “Cambiar clave” de la propia ventana.
5. Si el proceso ha finalizado de forma correcta aparecerá una ventana de confirmación.

3. COMPOSICIÓN

Para diseñar su propia composición fotográfica puede agregar diversos elementos: un escenario, personajes, complementos, imágenes propias. También puede realizar ajustes sobre los propios elementos y retoques fotográficos sobre la composición. Los cambios se verán reflejados en el panel proyecto que aparece sobre el panel de composición.

Para realizar las modificaciones y observar los cambios debe tener visible el panel de composición. Acceda al panel de composición mediante el submenú “Herramientas” situado en el menú superior, seleccione “Mostrar/Ocultar” y después “Composición”.



CAPÍTULO 4: MANUAL DE USUARIO

También puede acceder mediante el botón  situado en la barra de herramientas o la combinación de teclas “Ctrl+D”.

3.1. Escenarios

Los escenarios disponibles en la aplicación se muestran en la parte inferior del panel de composición (panel de imágenes) tras presionar sobre la pestaña “Escenarios”.

Búsquedas entre los escenarios disponibles:

1. Vaya al panel de imágenes situado en la parte inferior del panel de composición y seleccione la pestaña “Escenarios”.
2. Escriba en la casilla situada a la derecha de “Buscar” algún término relacionado con el escenario que desea encontrar.
3. Los personajes mostrados en el panel imágenes cambian según sus patrones de búsqueda introducidos.

Para mostrar información sobre uno de los escenarios haga “click” sobre la imagen de uno de ellos. En la parte derecha del panel de imágenes aparecerá su nombre completo y su descripción. Para más información sobre las características y recomendaciones del escenario presione sobre el botón “Información”.

Puede añadir un escenario a su composición haciendo “doble click” sobre la imagen de uno de ellos. Solo es posible añadir un único escenario a la composición, por lo tanto el escenario se sustituirá en la composición si ya había añadido uno previamente.

3.2. Personajes

Los personajes disponibles en la aplicación se muestran en la parte inferior del panel de composición (panel de imágenes) tras presionar sobre la pestaña “Personajes”.

Búsquedas entre los personajes disponibles:

1. Vaya al panel de imágenes situado en la parte inferior del panel de composición y seleccione la pestaña “Personajes”.
2. Escriba en la casilla situada a la derecha de “Buscar” algún término relacionado con el personaje que desea encontrar.



CAPÍTULO 4: MANUAL DE USUARIO

3. Seleccionar un tipo de personaje mediante los botones situados debajo del cuadro de búsqueda: “Todos”, “Hombres”, “Mujeres”, “Niños” y “Parejas”.
4. Los personajes mostrados en el panel imágenes cambian según sus patrones de búsqueda introducidos.

Para mostrar información sobre uno de los personajes haga “click” sobre la imagen de uno de ellos. En la parte derecha del panel de imágenes aparecerá su nombre completo y su descripción.

Añadir un nuevo personaje a su composición:

1. Vaya al panel de imágenes situado en la parte inferior del panel de composición y seleccione la pestaña “Personajes”. A continuación se mostrarán los personajes disponibles.
2. Puede realizar una búsqueda de personajes si lo necesita.
3. Seleccione un personaje mediante un “doble click” en el panel de imágenes.
4. Su personaje seleccionado aparecerá en el panel de edición (situado a la derecha del programa) tras presionar en la pestaña “Composición”.
5. Para confirmar que desea añadir el personaje presione sobre el botón “Añadir elemento” que aparece situado bajo la imagen de su personaje en el panel de edición.
6. El personaje aparecerá en su composición situado sobre la esquina superior izquierda en el primer plano de la composición.

Puede realizar ediciones sobre cada uno de los personajes añadidos a la composición.

Modificar la posición de un personaje:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque el personaje que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Modifique la posición en el eje X mediante el primero de los sliders situado junto al elemento que desea modificar.
5. Modifique la posición en el eje Y mediante el segundo de los sliders situado junto al elemento que desea modificar.



CAPÍTULO 4: MANUAL DE USUARIO

6. Su composición se verá modificada según estos controles, alterándose la posición del personaje correspondiente.

Modificar el tamaño de un personaje:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque el personaje que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Modifique el tamaño mediante el tercero de los sliders situado junto al elemento que desea modificar.
5. Su composición se verá modificada según estos controles, alterándose el tamaño del personaje correspondiente.

Cambiar de plano un personaje:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque el personaje que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Presionar el botón “Enviar atrás” situado junto al elemento que desea modificar para desplazar su personaje un plano en dirección al fondo de su composición.
5. Presionar el botón “Traer adelante” situado junto al elemento que desea modificar para desplazar su personaje un plano en dirección al frente de su composición.
6. Presionar el botón “Enviar al fondo” situado junto al elemento que desea modificar para desplazar su personaje al fondo de su composición (último plano de la composición).
7. Presionar el botón “Traer al frente” situado junto al elemento que desea modificar para desplazar su personaje al frente de su composición (primer plano de la composición).
8. Su composición se verá modificada según estos controles, alterándose los planos correspondientes.

Voltear horizontalmente un personaje:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.



CAPÍTULO 4: MANUAL DE USUARIO

3. Busque el personaje que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Presione sobre el botón “Espejo” situado junto al elemento que desea modificar.
5. Su composición se verá modificada según este control, alterándose la orientación horizontal del personaje correspondiente.

Eliminar un personaje:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque el personaje que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Presione sobre el botón “Eliminar” situado junto al elemento que desea modificar.
5. Su composición se verá modificada eliminándose de ella el personaje seleccionado.

3.3. Complementos

Los complementos disponibles en la aplicación se muestran en la parte inferior del panel de composición (panel de imágenes) tras presionar sobre la pestaña “Complementos”.

Búsquedas entre los complementos disponibles:

1. Vaya al panel de imágenes situado en la parte inferior del panel de composición y seleccione la pestaña “Complementos”.
2. Escriba en la casilla situada a la derecha de “Buscar” algún término relacionado con el complemento que desea encontrar.
3. Los complementos mostrados en el panel imágenes cambian según sus patrones de búsqueda introducidos.

Para mostrar información sobre uno de los complementos haga “click” sobre la imagen de uno de ellos. En la parte derecha del panel de imágenes aparecerá su nombre completo y su descripción.

Añadir un nuevo complemento a su composición:



CAPÍTULO 4: MANUAL DE USUARIO

1. Vaya al panel de imágenes situado en la parte inferior del panel de composición y seleccione la pestaña “Complementos”. A continuación se mostrarán los complementos disponibles.
2. Puede realizar una búsqueda de complementos si lo necesita.
3. Seleccione un complemento mediante un “doble click” en el panel de imágenes.
4. Su complemento seleccionado aparecerá en el panel de edición (situado a la derecha del programa) tras presionar en la pestaña “Composición”.
5. Para confirmar que desea añadir el complemento presione sobre el botón “Añadir elemento” que aparece situado bajo la imagen de su complemento en el panel de edición.
6. El complemento aparecerá en su composición situado sobre la esquina superior izquierda en el primer plano de la composición.

Puede realizar ediciones sobre cada uno de los complementos añadidos a la composición.

Modificar la posición de un complemento:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque el complemento que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Modifique la posición en el eje X mediante el primero de los sliders situado junto al elemento que desea modificar.
5. Modifique la posición en el eje Y mediante el segundo de los sliders situado junto al elemento que desea modificar.
6. Su composición se verá modificada según estos controles, alterándose la posición del complemento correspondiente.

Modificar el tamaño de un complemento:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque el complemento que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Modifique el tamaño mediante el tercero de los sliders situado junto al elemento que desea modificar.



CAPÍTULO 4: MANUAL DE USUARIO

5. Su composición se verá modificada según estos controles, alterándose el tamaño del complemento correspondiente.

Cambiar de plano un complemento:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque el complemento que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Presionar el botón “Enviar atrás” situado junto al elemento que desea modificar para desplazar su complemento un plano en dirección al fondo de su composición.
5. Presionar el botón “Traer adelante” situado junto al elemento que desea modificar para desplazar su complemento un plano en dirección al frente de su composición.
6. Presionar el botón “Enviar al fondo” situado junto al elemento que desea modificar para desplazar su complemento al fondo de su composición (último plano de la composición).
7. Presionar el botón “Traer al frente” situado junto al elemento que desea modificar para desplazar su complemento al frente de su composición (primer plano de la composición).
8. Su composición se verá modificada según estos controles, alterándose los planos correspondientes.

Voltear horizontalmente un complemento:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque el complemento que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Presione sobre el botón “Espejo” situado junto al elemento que desea modificar.
5. Su composición se verá modificada según este control, alterándose la orientación horizontal del complemento correspondiente.

Eliminar un complemento:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.



CAPÍTULO 4: MANUAL DE USUARIO

3. Busque el complemento que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Presione sobre el botón “Eliminar” situado junto al elemento que desea modificar.
5. Su composición se verá modificada eliminándose de ella el complemento seleccionado.

3.4. Imágenes propias

Añadir su propia imagen a la composición:

1. Presione sobre la pestaña “Composición” en el panel de edición (situado a la derecha del programa).
2. Arrastre una imagen en formato PNG disponible en su ordenador a la parte superior derecha del panel de edición, sobre el botón “Añadir elemento”.
3. Su imagen aparecerá ahora en dicha posición.
4. Para confirmar que desea añadir su imagen presione sobre el botón “Añadir elemento” que aparece situado bajo su imagen en el panel de edición.
5. Su imagen aparecerá en su composición situado sobre la esquina superior izquierda en el primer plano de la composición.

Puede realizar ediciones sobre cada uno de sus imágenes añadidas a la composición.

Modificar la posición de una imagen:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque la imagen que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Modifique la posición en el eje X mediante el primero de los sliders situado junto a la imagen que desea modificar.
5. Modifique la posición en el eje Y mediante el segundo de los sliders situado junto a la imagen que desea modificar.
6. Su composición se verá modificada según estos controles, alterándose la posición de su imagen correspondiente.



CAPÍTULO 4: MANUAL DE USUARIO

Modificar el tamaño de una imagen:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque la imagen que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Modifique el tamaño mediante el tercero de los sliders situado junto al elemento que desea modificar.
5. Su composición se verá modificada según estos controles, alterándose el tamaño de su imagen correspondiente.

Cambiar de plano una imagen:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque la imagen que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Presionar el botón “Enviar atrás” situado junto al elemento que desea modificar para desplazar su imagen un plano en dirección al fondo de su composición.
5. Presionar el botón “Traer adelante” situado junto al elemento que desea modificar para desplazar su imagen un plano en dirección al frente de su composición.
6. Presionar el botón “Enviar al fondo” situado junto al elemento que desea modificar para desplazar su imagen al fondo de su composición (último plano de la composición).
7. Presionar el botón “Traer al frente” situado junto al elemento que desea modificar para desplazar su imagen al frente de su composición (primer plano de la composición).
8. Su composición se verá modificada según estos controles, alterándose los planos correspondientes.

Voltear horizontalmente una imagen:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque la imagen que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.



CAPÍTULO 4: MANUAL DE USUARIO

4. Presione sobre el botón “Espejo” situado junto al elemento que desea modificar.
5. Su composición se verá modificada según este control, alterándose la orientación horizontal de su imagen correspondiente.

Eliminar una de sus imágenes:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Elementos” del mismo panel.
3. Busque la imagen que desea modificar en la lista de elementos de la composición mediante las imágenes situadas a la izquierda de cada uno de los elementos.
4. Presione sobre el botón “Eliminar” situado junto al elemento que desea modificar.
5. Su composición se verá modificada eliminándose de ella la imagen seleccionada.

3.5. Composición

Ajustar el brillo de la composición:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Composición” del mismo panel y diríjase a la parte inferior “Efectos y filtros”.
3. Desplace el primer *slider*, situado a la derecha de “Brillo”, para ajustar el nivel de brillo de su composición entre los valores -100 y 100.
4. El nivel de brillo sobre su composición se verá modificado según este control.

Ajustar el contraste de la composición:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Composición” del mismo panel y diríjase a la parte inferior “Efectos y filtros”.
3. Desplace el segundo *slider*, situado a la derecha de “Contraste”, para ajustar el nivel de contraste de su composición entre los valores -100 y 100.
4. El nivel de contraste sobre su composición se verá modificado según este control.



CAPÍTULO 4: MANUAL DE USUARIO





Ajustar el nivel de color de la composición:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Composición” del mismo panel y diríjase a la parte inferior “Efectos y filtros”.
3. Desplace el tercer *slider*, situado a la derecha de “Color”, para ajustar el nivel de color de su composición entre los valores -100 y 100.
4. El nivel de color sobre su composición se verá modificado según este control.

Ajustar la nitidez de la composición:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Composición” del mismo panel y diríjase a la parte inferior “Efectos y filtros”.
3. Desplace el cuarto *slider*, situado a la derecha de “Nitidez”, para ajustar el nivel de nitidez de su composición entre los valores -100 y 100.
4. El nivel de nitidez sobre su composición se verá modificado según este control.





Voltear horizontalmente la composición:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Composición” del mismo panel y diríjase a la parte inferior “Efectos y filtros”.
3. Presione sobre el botón  para activar el volteo horizontal de la composición (espejo). El botón de volteo horizontal pasará a estado “activado” .
4. Presione sobre el botón  para desactivar el volteo horizontal de la composición. El botón de volteo horizontal pasará a estado “desactivado” .
5. El volteo horizontal sobre la composición será efectuado inmediatamente.





Aplicar filtro “Posterizar” a la composición:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Composición” del mismo panel y diríjase a la parte inferior “Efectos y filtros”.





CAPÍTULO 4: MANUAL DE USUARIO

3. Presione sobre el botón  para activar el filtro de “posterización” sobre toda la composición. El botón pasará a estado “activado” .
4. Presione sobre el botón  para desactivar el filtro de “posterización”. El botón pasará a estado “desactivado” .
5. El filtro será aplicado sobre la composición inmediatamente.


Aplicar filtro “Equalizar” a la composición:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Composición” del mismo panel y diríjase a la parte inferior “Efectos y filtros”.
3. Presione sobre el botón  para activar el filtro de “equalización” sobre toda la composición. El botón pasará a estado “activado” .
4. Presione sobre el botón  para desactivar el filtro de “equalización”. El botón pasará a estado “desactivado” .
5. El filtro será aplicado sobre la composición inmediatamente.

Aplicar filtro “Alisar” a la composición:

1. Vaya al panel de edición (situado a la derecha del programa).
2. Seleccione la pestaña “Composición” del mismo panel y diríjase a la parte inferior “Efectos y filtros”.
3. Presione sobre el botón  para activar el filtro de “alisado” sobre toda la composición. El botón pasará a estado “activado” .
4. Presione sobre el botón  para desactivar el filtro de “alisado”. El botón pasará a estado “desactivado” .
5. El filtro será aplicado sobre la composición inmediatamente.

Exportar una composición creada:

1. Acceda a la ventana de exportación mediante el submenú “Proyecto” situado en el menú superior, y después seleccione “Exportar composición”. También puede acceder mediante el botón  situado la barra de herramientas o la combinación de teclas “Ctrl+M”.
2. Seleccione un directorio de su ordenador mediante las opciones ofrecidas en la ventana de “Exportar composición”.




CAPÍTULO 4: MANUAL DE USUARIO


3. Introduzca un nombre para el archivo de imagen que contendrá su composición en la casilla “Nombre” o elija uno de los archivos disponibles en el directorio seleccionado.
4. Seleccione el formato de archivo de imagen para su exportación en “Tipo”.
5. Presione sobre el botón “Guardar”.
6. El archivo guardado se abrirá automáticamente para que pueda ver el resultado final.
7. Acceda mediante su ordenador al directorio que seleccionó anteriormente para visualizar su archivo.

4. PROYECTO

Nuevo proyecto:

1. Acceda a la ventana “Nuevo proyecto” mediante el submenú “Proyecto” situado en el menú superior, y después seleccione “Nuevo”. También puede acceder mediante el botón  situado en la barra de herramientas o la combinación de teclas “Ctrl+N”.
2. Confirme abrir un nuevo proyecto en blanco mediante el botón “Si” de la nueva ventana.


Guardar proyecto:

1. Acceda a la ventana “Guardar proyecto” mediante el submenú “Proyecto” situado en el menú superior, y después seleccione “Guardar”. También puede acceder mediante el botón  situado en la barra de herramientas o la combinación de teclas “Ctrl+S”.
2. Seleccione un directorio de su ordenador mediante las opciones ofrecidas en la ventana de “Guardar proyecto”.
3. Introduzca un nombre para el archivo de guardado que contendrá su proyecto en la casilla “Nombre” o elija uno de los archivos disponibles en el directorio seleccionado.
4. Presione sobre el botón “Guardar”.
5. El archivo de guardado de su proyecto se encontrará en el directorio de su ordenador seleccionado anteriormente.




CAPÍTULO 4: MANUAL DE USUARIO

Abrir proyecto:

1. Acceda a la ventana “Abrir proyecto” mediante el submenú “Proyecto” situado en el menú superior, y después seleccione “Abrir”. También puede acceder mediante el botón  situado en la barra de herramientas o la combinación de teclas “Ctrl+O”.
2. Confirme la apertura de un nuevo proyecto perdiendo los cambios no guardados presionando el botón “Sí”.
3. Seleccione un directorio de su ordenador mediante las opciones ofrecidas en la ventana de “Abrir proyecto”.
4. Introduzca un nombre de un archivo de guardado de proyecto en la casilla “Nombre” o elija uno de los archivos disponibles en el directorio seleccionado.
5. Presione sobre el botón “Abrir”.
6. El proyecto se abrirá cargando su composición, elementos, efectos y filtros sobre la aplicación.


Abrir proyecto desde el panel de usuario:

1. Acceda al panel de usuario mediante el submenú “Herramientas” situado en el menú superior, seleccione “Mostrar/Ocultar” y después “Usuario”. También puede acceder mediante el botón  situado en la barra de herramientas o la combinación de teclas “Ctrl+U”.
2. En la parte derecha del panel de usuario aparecerán sus proyectos guardados en la carpeta predefinida por el programa. Se muestra en nombre de cada proyecto junto con una imagen en miniatura de su composición.
3. Haga “doble click” sobre las imágenes de composición de uno de los proyectos.
4. Confirme la apertura de un nuevo proyecto perdiendo los cambios no guardados presionando el botón “Sí”.
5. El correspondiente proyecto se abrirá cargando su composición, elementos, efectos y filtros sobre la aplicación.




5. OTROS


Enviar un correo al administrador de la aplicación:

1. Acceda a la ventana “Enviar correo” mediante el submenú “Herramientas” situado en el menú superior, y después seleccione “Enviar un correo”. También puede acceder mediante el botón  situado en la barra de herramientas o la combinación de teclas “Ctrl+E”.
2. Rellene todos los campos de correo remitente, asunto y mensaje en el formulario de la ventana “Enviar correo”. Los campos de correo remitente y asunto son obligatorios.
3. Pulse el botón “Enviar” de la propia ventana.
4. Si el correo se ha enviado de forma satisfactoria al administrador de la aplicación aparecerá una ventana de confirmación.

Acceder al área de administración:

1. Para acceder al área de administración de la aplicación puede hacerlo mediante el submenú “Herramientas” situado en el menú superior, y después seleccione “Administración”. También puede acceder mediante el botón  situado en la barra de herramientas o la combinación de teclas “Ctrl+A”.
2. Si tiene permisos de administración el área de administración se abrirá inmediatamente. Si no tiene permisos no podrá acceder a dicho área.

Editar las preferencias:

1. Acceda a la ventana “Preferencias” mediante el submenú “Herramientas” situado en el menú superior, y después seleccione “Preferencias”. También puede acceder mediante el botón  situado en la barra de herramientas o la combinación de teclas “Ctrl+P”.
2. Seleccione la casilla “Guardar la vista al cerrar el programa” para permitir al programa recordar la configuración visual del programa en el momento de cierre y recordarla en las próximas aperturas del programa. Deseleccione la casilla para desactivar esta opción.
3. Seleccione la casilla “Abrir último proyecto guardado al iniciar el programa” para permitir al programa recordar cuál ha sido el último proyecto editado y así cargarlo directamente en la próximas apertura del programa. Deseleccione la casilla para desactivar esta opción.



CAPÍTULO 4: MANUAL DE USUARIO

4. Seleccione la casilla “Guardar el usuario al cerrar el programa” para permitir al programa cargar sus datos de usuario y clave en la pantalla de inicio en las próximas aperturas del programa. Deseleccione la casilla para desactivar esta opción.
5. Elija mediante los botones disponibles en la misma ventana un color para la aplicación. Las posibilidades son distintas tonalidades de grises: “Claro”, “Medio” y “Oscuro”. Reinicie la aplicación para observar los cambios.
6. Presione el botón “Ok” para guardar los cambios en la configuración de la aplicación y “Cancelar” si no desea guardar las modificaciones realizadas.

Consultar información sobre la aplicación:

1. Si desea consultar información sobre la aplicación que está utilizando vaya al submenú “Ayuda” situado en el menú superior, y después seleccione “Información”. También puede acceder mediante la combinación de teclas “Ctrl+I”.
2. Despliegue los distintos tipos de información mediante los botones de la ventana “About Autodesign”.

Abrir el manual de usuario:

Puede acceder directamente al presente manual de usuario desde la propia aplicación.

1. Para acceder al manual de usuario puede hacerlo mediante el submenú “Ayuda” situado en el menú superior, y después seleccione “Manual de usuario”. También puede acceder mediante la combinación de teclas “Ctrl+H”.

Cambiar de usuario:

Puede realizar un cambio de usuario en la aplicación sin necesidad de cerrar el propio programa.

1. Para cambiar de usuario puede hacerlo mediante el submenú “Salir” situado en el menú superior, y después seleccione “Cambiar de usuario”. También puede acceder mediante la combinación de teclas “Ctrl+Q”.
2. Rellene los campos de usuario y clave de la ventana de inicio con los datos de correspondientes de su registro.
3. Si alguno de los campos no se ha introducido correctamente siga las instrucciones que aparecen sobre la propia ventana.
4. Pulse el botón “Entrar” de la propia ventana de inicio.



CAPÍTULO 4: MANUAL DE USUARIO

5. Si el proceso ha finalizado de forma correcta aparecerá la ventana principal de la aplicación.

Cerrar la aplicación:

1. Para cerrar el programa hacerlo mediante el submenú “Salir” situado en el menú superior, y después seleccione “Cerrar”. También puede cerrar mediante el botón superior derecho de la ventana de inicio, registro o principal o mediante la tecla “Esc”.
2. Confirme el cierre del programa perdiendo los cambios no guardados en el proyecto.



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio dedicada a la composición fotográfica: Autodesign

CAPÍTULO 4: MANUAL DE USUARIO



CAPÍTULO 5: ANÁLISIS FINAL

1. PRESUPUESTO ECONÓMICO

Cualquier tipo de desarrollo software a medida, dedicado a cumplir requisitos particulares y específicos fijados por un cliente, supone un precio a pagar. En consecuencia, se elabora un presupuesto aproximado y orientativo que ayude a aproximarnos a una situación laboral real, ya que la aplicación desarrollada en este proyecto no está realizada con fines lucrativos.

Se consideran los precios atendiendo al trabajo realizado hasta ahora, es decir, a la primera versión de la aplicación desarrollada en este proyecto. El mayor porcentaje de los costes en el desarrollo software se basan en la cantidad de trabajo o tiempo de desarrollo que se requiere para entregar el proyecto según las especificaciones del cliente, y en la complejidad de dichas especificaciones. A mayores, existen otros factores que pueden hacer oscilar el precio final del software, pero en menor medida.

El tiempo de desarrollo del trabajo viene dado en mayor parte por el tiempo dedicado a la programación del propio software. En algunas empresas utilizan alternativamente como medida el número de líneas de código, aunque desde mi punto de vista me parece una opción menos adecuada. Por lo tanto, el cálculo de este presupuesto se realizará en base al tiempo empleado en el desarrollo del proyecto. El análisis de requerimientos, organización del propio proyecto, diseño gráfico y estético de la interfaz y los métodos de instalación son, entre otras, labores que también han de tenerse en cuenta en el cómputo del tiempo de desarrollo. Basándonos en estos conceptos, se estima un tiempo aproximado de 250 horas de trabajo, excluyendo de forma estimada parte del tiempo dedicada al aprendizaje. Basándome en mi escasa experiencia de programación en el mundo laboral se utiliza un sueldo de programador junior en España con baja experiencia para el cálculo del presupuesto. Éste es de aproximadamente 13.000€ anuales. Con los datos propuestos se estima un coste para este apartado de 1.540€.

El precio determinado para una aplicación software es influido, en parte, por la complejidad del proyecto a desarrollar. Proyectos que requieran de características y funcionalidades más complejas supondrán, como es lógico, precios más elevados. Se consideran de perfil básico la mayor parte de las funcionalidades de la aplicación en



CAPÍTULO 5: ANÁLISIS FINALES

cuanto a complejidad se refiere, y por lo tanto no suponen costes adicionales. La funcionalidad de composición de los proyectos a partir de múltiples imágenes requiere un trato diferente, ya que se considera algo más compleja y se estima un coste adicional de 500€ (basando la estimación de este dato en un presupuesto real de la aplicación).

No se incluyen tampoco costes debidos al software utilizado para el desarrollo del proyecto, ya que los programas que han intervenido (Eclipse, XAMPP...) son de licencia gratuita.

Contribuyen también a elevar el coste de un proyecto la urgencia requerida por parte de un cliente, ya que se puede considerar como un requisito extra. A pesar de tener fijada una fecha límite para la entrega del TFG, no se considera que la intensidad de trabajo se haya visto incrementada debido a este hecho, ya que no se ha trabajado bajo unos objetivos estrictamente rígidos y estos se han adaptado a medida que se avanzaba en el desarrollo. Por lo tanto el concepto de “urgencia” tampoco añadirá costes adicionales.

Para acercarnos algo más al precio de mercado real se tendrá en cuenta un margen de beneficio empresarial o de autónomo del 20% (dato a modo de ejemplo), aplicándolo correctamente sobre los costes calculados. Se incluirá también el concepto del IVA correspondiente al 21% del precio final calculado hasta el momento. De acuerdo a estos factores, el precio final de la aplicación sería de 3085€.

Además, para ampliar la información en torno al presupuesto de la aplicación, se ha solicitado también un presupuesto estimado en la organización dedicada al desarrollo software en la que se han realizado de forma paralela las prácticas en empresa correspondientes al Grado en Tecnologías de Telecomunicación. Fundación Aptiva es una organización orientada a acercar las herramientas TICs a las pequeñas empresas, y que está dando sus primeros pasos en el mundo del desarrollo software. En consecuencia suelen presentar presupuesto de carácter bastante más económico en relación a otras empresas con un perfil más profesional y con una mayor experiencia en este terreno. El presupuesto, aproximado de nuevo, ofrecido por esta organización ronda las cifras de los 2.500 € para la aplicación software realizada en este proyecto.

Como ya se ha comentado, todas las cifras manejadas son estimaciones imprecisas e inexactas, y en consecuencia no pueden ser tomadas como cifras reales. Pero sí nos conducen a tener una idea aproximada de lo que puede suponer el precio correspondiente al desarrollo de una aplicación de estas características. Para acabar, y como curiosidad, el proyecto completo de la aplicación Autodesign (no solo la versión 1.0 desarrollada en este proyecto) fue planteado en una empresa profesional dedicada



CAPÍTULO 5: ANÁLISIS FINALES

al desarrollo software. El presupuesto orientativo calculado por esta empresa fue de 15.000€ (incluyendo soporte para descarga y actualizaciones, gestión de subida de proyectos, métodos de pago y las funcionalidades futuras analizadas en el apartado anterior).

2. CONCLUSIONES

Se exponen a continuación los beneficios e ideas adquiridos que ha supuesto la realización del trabajo desde un punto de vista más personal. Se hace necesario mencionar que al término del trabajo y a nivel global, los resultados obtenidos superan ampliamente las expectativas iniciales, planteadas desde la incertidumbre que suponía realizar un Trabajo Fin de Grado.

Comenzando por el ámbito teórico, ha sido esencial “empaparse” y adentrarse en los campos intelectuales que rodean el escenario de trabajo, lo que me ha permitido adquirir gran cantidad de conocimientos incluso en áreas bien diferenciadas. La posibilidad de escoger inicialmente el proyecto a realizar, me ha permitido explorar campos teóricos que me suscitan cierto interés, lo que de verdad creo que ha fomentado en gran medida esta fase del trabajo.

Por una parte, en el ámbito de la imagen, he podido afianzar mis nociones básicas así como adquirir nuevos conceptos de forma sencilla. Por otro lado, la investigación relacionada con las tecnologías a utilizar ha sido más profunda e intensa, pues es donde se han debido tomar decisiones más importantes de cara la elaboración de la aplicación. El carácter más técnico de este campo de estudio, repercute obviamente en una asimilación más lenta de los conceptos teóricos. Sin embargo, se han conseguido los objetivos en este apartado sin excesivas complicaciones. En parte, debido a su mayor relación con los contenidos vistos en la carrera, con respecto a los otros ámbitos de estudio. Las asignaturas “Programación”, “Arquitectura de Redes, Sistemas y Servicios”, “Ingeniería de Sistemas Software”, “Desarrollo de Aplicaciones Distribuidas”, “Administración y Gestión de Redes y Servicios Telemáticos” y, por encima de estas, “Laboratorio de Desarrollo de Sistemas Telemáticos”, propias de Grado en Ingeniería de Tecnologías de Telecomunicación, se consideran las más influyentes en este apartado. En tercer lugar, el estudio relacionado con el software, ha sido menos exhaustivo, centrándose en los conceptos puntuales más vinculados al proyecto. Aunque no por ello, ha sido menos importante.



CAPÍTULO 5: ANÁLISIS FINALES

El proceso de documentación llevado a cabo, se ha basado principalmente en la información encontrada en Internet. La información que se puede encontrar vía web es descomunadamente extensa, más aún en los campos tecnológicos estudiados que sufren un cambio permanente. Pero precisamente por ello, es necesario llevar a cabo un importante proceso de selección, tanto a nivel de contenidos como a nivel de veracidad. Mis avances en este aspecto han sido notables. El perfeccionamiento de esta búsqueda de información se impulsa mediante la práctica, mediante la cual se adquieren metodologías y e información de aquellos lugares donde poder extraer la mejor información posible de la forma más directa.

Prestando atención al apartado práctico realizado, los avances conseguidos han sido innumerables. He comprendido como una buena planificación y metodología son imprescindibles para el desarrollo de un buen proyecto. Con esto no quiero decir que estos conceptos hayan sido estrictamente los más correctos, sino que los he comprendido en base al análisis de los fallos cometidos. De la misma forma, un correcto análisis teórico influye en la toma de decisiones de cara a la fase de desarrollo técnico. El tiempo estimado inicialmente para el desarrollo de la aplicación, se ha visto excedido de forma notable, de lo que se deduce que el apartado de planificación no ha sido el más acertado.

La fase de desarrollo se ha visto fuertemente influenciada por una de las primeras decisiones tomadas: el aprendizaje de un nuevo lenguaje de programación. Como ya se ha comentado a lo largo del trabajo, la elección fue el lenguaje Python. Obviamente, introducirme desde cero en su aprendizaje ha supuesto ser uno de los puntos donde mis conocimientos teóricos y prácticos se han incrementado de forma más notoria. Aprovecho para mencionar, que Python ha cumplido perfectamente con lo que se esperaba de él, y lo considero altamente recomendable para el desarrollo de aplicaciones de escritorio. Ya no se habla solo del propio lenguaje, sino la posibilidad de ver el desarrollo software desde un nuevo punto de vista. Finalizado el proyecto, puedo decir que estoy realmente satisfecho con los avances conseguidos en este apartado. Durante esta etapa, he conseguido desarrollar en gran medida métodos de auto-aprendizaje. Acostumbrado a realizar este tipo de proyectos de forma más colectiva, me he visto forzado a solucionar la totalidad de las adversidades técnicas surgidas por mí mismo, lo que valoro de forma muy positiva.

El diseño de la interfaz gráfica de la aplicación, relativamente compleja respecto a mi nivel inicial, ha supuesto las mayores complicaciones durante esta fase de desarrollo. El origen de estas dificultades surge a raíz de, por un lado, el desconocimiento del lenguaje de programación, y por otro, la inexperiencia en este sector de la



CAPÍTULO 5: ANÁLISIS FINALES

programación. De nuevo, el resultado final me satisface plenamente. Otro de los puntos clave, ha sido el desarrollo de la capacidad para ponerse en el papel de un usuario, a la hora de realizar cualquier tipo de software. Esta es una habilidad que, desde mi punto de vista, cobra bastante importancia de cara a perfeccionar el resultado final. En cuanto a los objetivos propuestos inicialmente, considero que se han cumplido satisfactoriamente, teniendo en cuenta que tener objetivos “abiertos” o variables permite conseguir mejores resultados.

A nivel personal, la experiencia que ha supuesto la realización del trabajo se considera positiva, ya que ha supuesto importantes beneficios. Considero el TFG como una pequeña etapa a medio camino entre la formación del Grado y el mundo laboral, lo que junto a las prácticas en empresa realizadas, han supuesto una experiencia realmente provechosa de cara al futuro. Las prácticas realizadas de forma paralela han ayudado de forma considerable. No a tanto nivel de programación, ya que no se hayan realizado labores estrechamente relacionadas, pero si a un nivel superior, el de desarrollo software. Mi interpretación de este concepto ha cambiado para bien, valorando desde dentro el esfuerzo, valor e incluso precio que supone realizar productos de este carácter.

Uno de los puntos que más valoro de la experiencia conjunta de ambas, Trabajo Fin de Grado y prácticas en empresa, es que me han permitido descubrir algunos de mis puntos fuertes y débiles a varios niveles. Este hecho adquiere una gran importancia, si con ello se es capaz de aprovechar esas habilidades o capacidades destacadas y de mejorar en aquellos aspectos que realizamos con más complicación.

Se espera del producto desarrollado que pueda satisfacer los requisitos técnicos impuestos en la fase de iniciación del proyecto. Se pretende también con esta aplicación haber dado un primer paso en el acercamiento de la edición digital de imagen al posible público menos relacionado con este ámbito. Como se ha mencionado en otros apartados del documento, se ha considerado de vital importancia ofrecer una grata experiencia de usuario con la intención de que la totalidad de usuarios pueda extraer los beneficios de esta aplicación. Fomentar la creatividad y la expresión artística, son también aspectos clave en los objetivos de la aplicación. Por otro lado, se ha considerado importante conseguir una buena base de cara a las posibles versiones futuras de la aplicación. A nivel de programación, la estructuración y claridad de código se consideran factores muy influyentes en este aspecto. Se han visto reforzados muy positivamente gracias al lenguaje de programación Python, el cual recomiendo ya que tiene como característica propia conseguir precisamente estos objetivos. Conseguir finalizar favorablemente algunos de



CAPÍTULO 5: ANÁLISIS FINALES

los objetivos propuestos, aunque no hayan sido estrictamente todos, supondrá un buen punto de arranque para el desarrollo de la siguiente versión. Sería muy interesante, de cara al futuro de la aplicación, realizar un encuesta de opinión a un variado público, sobre la aplicación planteada hasta el momento. Esto permitiría reforzar e incidir en aquellos puntos que se consideran más débiles según los usuarios.

Finalmente, se considera la experiencia sobre el trabajo realizado muy positiva. Lograr satisfacer los objetivos, superando adversidades y dificultades, me han permitido avanzar en aspectos técnicos, intelectuales y personales. Es interesante mencionar, que en caso una ficticia repetición del trabajo o la realización de uno de características similares, abordaría el mismo siguiendo metodologías completamente diferentes. De esto se deduce, que he podido mejorar en los distintos campos que abarcan la totalidad del proyecto, en base al descubrimiento y corrección de mis propios fallos y defectos.

3. LÍNEAS FUTURAS

La aplicación diseñada en este proyecto constituye una primera versión de la aplicación: Autodesign 1.0. Por lo tanto, ésta supone un primer paso para la creación de la herramienta completa.

La complejidad que supone desarrollar por completo la aplicación “Autodesign” implica que aún quede mucho trabajo por hacer a partir de esta versión 1.0. A continuación se exponen una serie de ideas para llevar a cabo en el desarrollo de las posteriores versiones de la aplicación. Algunas forman parte de las características finales que tendría el programa y que no se han implementado en esta primera versión. Otras son simples mejoras de la versión propuesta en este proyecto. También se incluyen algunas propuestas para mejorar la usabilidad de la aplicación y se plantean algunos puntos donde conviene solucionar posibles errores de la aplicación o mejorar algunas características.

3.1. Mejoras de carácter general

Extensión de la aplicación a las plataformas de otros sistemas operativos.

Como ya se ha mencionado anteriormente, la aplicación ha sido desarrollada para el funcionamiento exclusivamente en sistemas operativos Windows. De hecho, el archivo ejecutable del propio programa, “Autodesign.exe”, solo es válido para sistemas Windows ya que, en principio, son los únicos que trabajan



CAPÍTULO 5: ANÁLISIS FINALES

con archivos en este formato. Sería importante de cara al futuro extender la aplicación a otros sistemas operativos, Linux y Mac preferentemente, para así poder abarcar un público más amplio.

Espacio en disco de la aplicación.

Los recursos de imagen utilizados en la aplicación suponen un importante aumento del tamaño de la aplicación. Se propone obtener una mejor relación de compresión en las imágenes utilizadas, de forma que se pueda reducir el peso de la aplicación. Pero para ello hay que tener en cuenta la reducción de calidad de las imágenes que esto supone, y llegar a una situación de compromiso que permita reducir el peso de la aplicación si afectar de forma apreciable la calidad de las imágenes dentro del programa.

Se podría valorar también la posibilidad de ofrecer a los usuarios otra versión de la aplicación con las imágenes almacenadas en un servidor y la descarga de estas de forma dinámica según su uso. Supondría una versión de peso muy reducido, pero con un peor rendimiento.

3.2. Solución de errores

Interfaz gráfica – Responsividad.

La interfaz gráfica presenta defectos en algunas situaciones críticas debido a su responsividad y a la carga dinámica de elementos gráficos. Convendría forzar la aplicación para llegar a las situaciones límite y mejorar este apartado.

Testeo en diferentes versiones Windows.

Se ha comprobado que la aplicación no responde siempre de la misma forma, dependiendo de la versión del sistema operativo utilizada, llegando en algunos casos a provocar errores. Es posible llevar a cabo mejoras en este punto descubriendo los fallos mediante pruebas en las distintas versiones del sistema operativo Windows.

Bloqueos cuando se produce pérdida de conexión a Internet

Otro apartado mejorable es la gestión de errores en la pérdida de conexión con la base de datos de la aplicación.



CAPÍTULO 5: ANÁLISIS FINALES

3.3. Mejoras de usabilidad

“Arrastrar y soltar”.

Implementar la posibilidad de llevar acciones mediante el método “arrastrar y soltar” (“drag and drop”) mediante el ratón supondría un importante impulso para la usabilidad de la aplicación. Introducir este sistema para la inclusión de escenarios, personajes y complementos en la composición mejoraría la facilidad de uso del programa.

“Ayuda en vivo”.

Se propone introducir un sistema de “ayuda en vivo”, mediante el cual se mostrarían mensajes al usuario a medida que crea su composición. Estos mensajes o consejos estarían relacionados con el manejo de la aplicación y la composición gráfica.

Accesibilidad por teclado.

La idea consiste en complementar el uso del ratón para el manejo de la aplicación con algunas funcionalidades controladas mediante el uso del teclado. Por ejemplo la navegación del panel de imágenes mediante el uso de las flechas del teclado, o el cambio de plano de los elementos mediante las teclas numéricas.

3.4. Mejoras en las funcionalidades existentes

Rendimiento de la aplicación.

Se considera el rendimiento en el procesado de la composición un apartado mejorable. En la actual versión, el programa procesa toda la composición cada vez que se realiza un cambio sobre ella. Esto supone tiempos de procesado algo elevados en algunos casos. Se proponen dos posibles ideas para mejorar este apartado de rendimiento de la aplicación:

- a) La primera idea consiste en utilizar imágenes de calidad reducida para el procesado. Se incluirían en la aplicación imágenes de menor resolución, y por lo tanto peor calidad, para el procesamiento de la composición en los proyectos de los usuarios. Por otro lado, se utilizarían imágenes de una mayor calidad para generar la composición final exportada, manteniendo así (o incrementando) la calidad en el resultado final. En contrapartida, el duplicado de imágenes puede incrementar en exceso el tamaño de la aplicación.



CAPÍTULO 5: ANÁLISIS FINALES

- b) La segunda idea consiste en separar el procesado de la composición en los distintos planos. Procesar solo aquellos planos en los que haya modificaciones agilizaría la aplicación en gran medida. Para ello sería necesario almacenar las distintas capas de imagen procesadas durante la ejecución del programa. Además, se podría extender esta metodológica para la vista de la composición. En lugar de mostrar el resultado completo de la composición procesada, se visualizarían las capas de imagen superpuestas, de forma que solo se actualizaría la vista de aquellas que hayan sido modificadas en un determinado momento. Es una idea compleja de implementar, pero a priori parece viable.

Actualización de los recursos de imagen.

Introducir nuevos escenarios, personajes y complementos aumentaría las posibilidades y libertad del usuario a la hora de desarrollar su composición. Además, permitiría a la funcionalidad del buscador adquirir una mayor importancia. Las nuevas imágenes se irían incluyendo en la aplicación de forma progresiva.

3.5. Incorporación de nuevas funcionalidades

Módulo de edición de fotografías: recortar, rotar y mover.

Se introduciría un módulo en el panel de edición que permita que editar tanto las imágenes de personajes y complementos como las imágenes importadas mediante las funciones básicas de edición: cortar, rotar y mover. Así como fusionar las imágenes importadas con los personajes propios de la aplicación. Se ha investigado sobre la posibilidad de implementar esta funcionalidad, y se ha deducido que es una idea bastante compleja de implementar.

Sistema de pedidos – proyectos – compras.

Otra funcionalidad de futuro, de cara a las próximas versiones de la aplicación es la idea de crear un sistema de gestión pedidos. Esto supondría implementar un sistema de compras en la propia aplicación, relacionadas directamente con los proyectos creados por los usuarios.

Área de administración.

Implementación de un área de administración de manejo sencillo vía web. El desarrollo se llevaría a cabo mediante Python y su *framework* para el desarrollo web Django.



CAPÍTULO 5: ANÁLISIS FINALES

Funcionalidades “Deshacer” y “Rehacer”.

Las conocidas funcionalidades “deshacer” y “rehacer” son muy útiles en gran cantidad de soluciones software, más aún en las dedicadas a la edición gráfica. Es posible llevar a cabo su implementación mediante el almacenamiento de una pila de acciones de usuario durante la ejecución de la aplicación. Aunque esta es también una idea especialmente compleja de desarrollar para este tipo de aplicación.

Para finalizar, en caso de llevarse a cabo la comercialización y distribución de la aplicación, sería conveniente implementar un soporte para la descarga de la aplicación desde una web. Y, además, un sistema de gestión de actualizaciones en la propia aplicación, permitiendo que esta se actualice a nuevas versiones y descargue los nuevos recursos de imagen en caso de que sea necesario. En caso de implementarse la funcionalidad de “compras” en el propio programa, sería necesario implementar el servicio de comercio electrónico mediante una pasarela de pago para las transacciones.



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio dedicada a la composición fotográfica: Autodesign

CAPÍTULO 5: ANÁLISIS FINALES



BIBLIOGRAFÍA

- andrearrrs. (2014). XML JSON YAML: Formatos para intercambiar información. Recuperado de <http://hipertextual.com/archivo/2014/05/xml-json-yaml/>.
- Arribas Á.G. (2014). Tipos de imágenes y formatos. Recuperado de <https://sites.google.com/site/ticvalcarcel/optimizacion-de-imagenes-para-internet/tipos-de-imagenes-y-formatos>.
- Avram A. (2011). Is REST Successful in the Enterprise? Recuperado de <http://www.infoq.com/news/2011/06/Is-REST-Successful>.
- Carrodegua, N. (2015). Como instalar y configurar el servidor web Apache en Windows. Recuperado de <http://norfipc.com/internet/instalar-servidor-apache.html>.
- Chau, C. (2014). REST vs SOAP. Recuperado de <http://mariochau.blogspot.com.es/2014/04/rest-vs-soap.html>.
- Cuarental Gazapo, C. (2015). INSTALACIÓN DE PYDEV EN ECLIPSE. Recuperado de <http://comiendobocabytes.com/instalacion-de-pydev-en-eclipse/>.
- Davidson, S.R. (2009). Curso de Gráficos: Conceptos. Recuperado de <http://graficos.conclase.net/curso/?cap=001>.
- González Duque, R. (2007). Python para todos. Recuperado de <http://edge.launchpad.net/improve-python-spanish-doc/0.4/0.4.0/+download/Python%20para%20todos.pdf>.
- Hernández, J.R. (2011). Formatos de imagen: Guía de optimización. Recuperado de <http://www.emezeta.com/articulos/formatos-de-imagen-guia-de-optimizacion>.
- Hernández, J.R. (2013). 15 aplicaciones para montar servidores web en local. Recuperado de <http://www.emezeta.com/articulos/15-aplicaciones-para-montar-servidores-web-en-local>.
- Hevíá, A. (2011). Implementación de SOA con REST. <http://pensandoensoa.com/2011/03/19/869/>.
- Kioskea. (2014). Lenguajes de programación. Recuperado de <http://es.kioskea.net/contents/304-lenguajes-de-programacion>.



BIBLIOGRAFÍA

Linblad, Thomas, Kinser J.M. (2013). *Image Processing using Pulse-Coupled Neural Networks*. Springer.

Luis, A. (2010). Herramientas (IDE, GUI, EDITOR) para desarrollar en Python. Recuperado de http://elviajedelnavegante.blogspot.com.es/2010/10/herramientas-ide-gui-editor-para_18.html.

Mateos, M. (2011). Formatos de imágenes: ¿cuál usar en cada caso? Recuperado de <http://www.genbeta.com/imagen-digital/formatos-de-imagenes-cual-usar-en-cada-caso>.

Mayta Flores, C.A. (2012). DEBATE ENTRE SERVICIOS WEB BASADOS EN REST Y SOAP. Recuperado de <http://carlosmayta.blogspot.com.es/>.

Moreno, J.M. (2010). Software web vs software “no web” o de escritorio (1/2). Recuperado de <http://www.prismasoftwaregestion.com/blog/software-web-vs-software-no-web-o-de-escritorio-12/>.

Moreno, J.M. (2011). Software web vs software “no web” o de escritorio (2/2). <http://www.prismasoftwaregestion.com/blog/software-web-vs-software-%E2%80%9Cno-web%E2%80%9D-o-de-escritorio-22/>.

overblog . (2011). Tipos De Bases De Datos. Recuperado de <http://basededatos.over-blog.net/article-tipos-de-bases-de-datos-68319538.html>.

PYPL. (2015). PYPL PopularitY of Programming Language. Recuperado de <http://pypl.github.io/PYPL.html>.

Python. (2014). Python2orPython3. Recuperado de <https://wiki.python.org/moin/Python2orPython3>.

Rodriguez Alonso, Hugo. (2013) *Imagen digital: conceptos básicos*. Barcelona, España: Marcombo.

Sierra, M. (2006). Qué es un servidor y cuáles son los principales tipos de servidores (proxy,dns, web,ftp,pop3 y smtp, dhcp...). Recuperado de http://aprenderaprogramar.es/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179.

Valls Arnau, M. (2011). Qué formato de imagen escoger. Recuperado de <https://mvallsa.wordpress.com/2014/02/06/que-formato-de-imagen-escoger/>



BIBLIOGRAFÍA

ZetCode. (2011). wxPython tutorial . Recuperado de <http://zetcode.com/wxpython/>



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio dedicada a la composición fotográfica: Autodesign

BIBLIOGRAFÍA



ANEXOS

1. INSTALACIÓN DEL LENGUAJE DE PROGRAMACIÓN: PYTHON

Python es uno de los posibles lenguajes de programación que podemos utilizar para desarrollar una aplicación de escritorio. Si nuestro objetivo es utilizar Python para desarrollar cualquier tipo de software el primer paso es instalarlo y configurarlo en la computadora en la que vayamos a trabajar. A continuación se muestran los pasos necesarios para un sistema operativo Windows:

1. Identificar la versión de Windows instalada en el ordenador.

En las propiedades de nuestro sistema podemos encontrar la edición de Windows que estamos utilizando (Windows Vista, Windows 7, Windows 8,...) y el tipo de sistema operativo que utilizamos (32 o 64 bits).

2. Decidir la versión de Python que queremos instalar.

Existen dos versiones de Python actualmente, Python 2 y Python 3. Sus versiones más actualizadas son Python 2.7.10 y Python 3.4.4, respectivamente. Python 3 es la versión más moderna y puede decirse es el futuro del propio lenguaje, mientras que Python 2 es más antiguo pero tiene la ventaja de ser un lenguaje mucho más completo. Mi recomendación es que si vas a empezar aprendiendo el lenguaje poco a poco desde 0 elijas Python 3, ya que es un lenguaje más nuevo y en crecimiento. Si por el contrario necesitas desarrollar ya mismo un programa, aplicación o herramienta y no sabes realmente a lo que te vas a enfrentar, elije Python 2, ya que es un lenguaje desarrollado por completo y no te va a dejar “tirado” prácticamente en ninguna situación. Para más información puedes consultar la siguiente web: <https://wiki.python.org/moin/Python2orPython3>.

3. Selecciona tu instalador en la página oficial de Python: www.python.org.

En base a las versiones de Windows y Python seleccionadas en los pasos previos descarga tu instalador de Python correspondiente. En la página web <https://www.python.org/downloads/windows/> tienes disponibles todas las últimas versiones. Haz “click” en el enlace de descarga elegido. Puedes descargar el archivo y luego ejecutarlo, o bien puedes ejecutarlo automáticamente. El proceso de instalación comenzará automáticamente.



4. Instalador de Python.

- 4.1. Ejecuta el instalador, si no lo ejecutaste automáticamente desde el diálogo de descarga. El programa de instalación se abrirá.
- 4.2. Haz "click" en "Instalar para todos los usuarios" si es necesario y luego en "Siguiete".
- 4.3. Escoge un directorio de instalación para Python. El valor por defecto, "C: \ Python27" (suponiendo que hayamos elegido Python2 como ha sido mi caso), pero puedes elegir otro si quieres. Recomiendo dejar el valor de directorio por defecto, ya que otras herramientas o extensiones de Python pueden venir configuradas por defecto para esta ruta. De esta forma, te puedes evitar algún problema o configuración extra. Sin embargo, si necesitas hacerlo en un directorio diferente por algún motivo, puedes hacerlo.
- 4.4. Elige las funciones que deseas instalar si es necesario y pulsa sobre "Siguiete" para iniciar la instalación.
- 4.5. Espera unos minutos para que el proceso se complete.
- 4.6. Una vez que se haya instalado Python, haz clic en "Finalizar" para cerrar el instalador.

5. Comprobar la instalación.

- 5.1. Abre el símbolo del sistema de Windows haciendo clic en "Inicio", escribe "cmd" en el cuadro de búsqueda y presiona "Aceptar".
- 5.2. Cambia a tu directorio de Python. Si aceptaste el directorio por defecto, escribe "cd C:\Python27" y presiona "Intro". Si lo has instalado en otro directorio, cambia "C:\Python27" por la ruta de tu directorio de instalación.
- 5.3. Escribe "python" y presiona "Enter" para iniciar la línea de comandos de Python del sistema.
- 5.4. Si el último comando "python" ha sido reconocido correctamente, la instalación se habrá completado con éxito.



Ilustración 32 – Comando “python”

En caso contrario, “python” no será reconocido como comando. Revisa la variable de entorno “Path” para comprobar que todo esté configurado correctamente.

Ilustración 33 – Comando “python” no reconocido

6. Configurar variables de entorno para Python en Windows.

Realizar esta configuración permite ejecutar *scripts* de Python sin necesidad de acceder al intérprete de Python “python.exe” ubicado en el directorio de instalación. Para ello debemos editar las variables de entorno de nuestro sistema operativo:

- 6.1. Acceder a “Inicio”.
- 6.2. Hacer clic derecho sobre “Equipo” que aparecerá en “Inicio” y seleccionar “Propiedades”.
- 6.3. En las propiedades de sistema elegir “Configuración avanzada del sistema”.
- 6.4. Pulsar en la parte de debajo de la ventana sobre el botón “Variables de entorno...”.

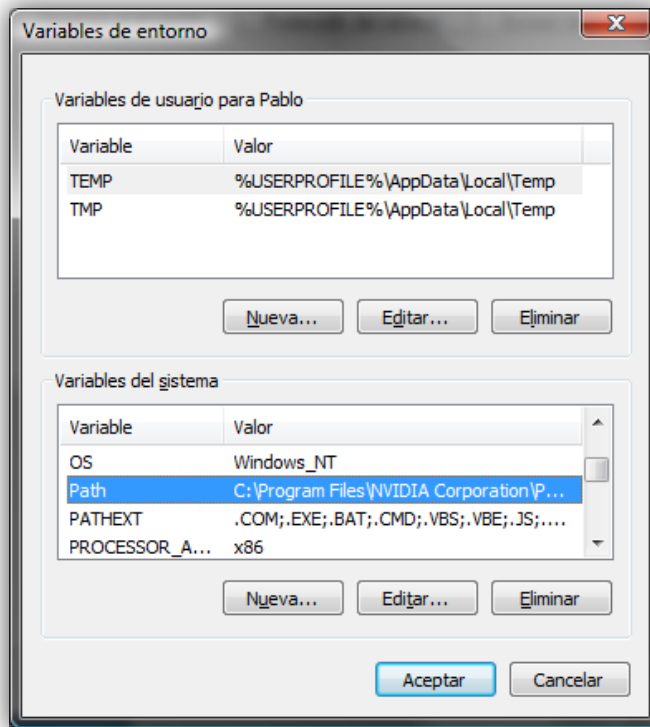


Ilustración 34 – Configuración de variables de entorno

- 6.5. En la sección de abajo “Variables del Sistema” buscar la variable “Path” y seleccionarla.
- 6.6. Presionar sobre el botón “Editar” para modificar la variable “Path”.
- 6.7. En la casilla “valor de la variable” debemos editar su valor añadiendo a continuación las siguientes rutas (nunca eliminar el valor de la variable o sustituir, simplemente añadir al final):
 - a) En caso de haber seleccionado la ruta por defecto:
C:\Python27;
C:\Python27\Scripts;
 - b) En caso de haber seleccionado una ruta diferente durante la instalación:
[ruta de instalación];
[ruta de instalación]\Scripts;

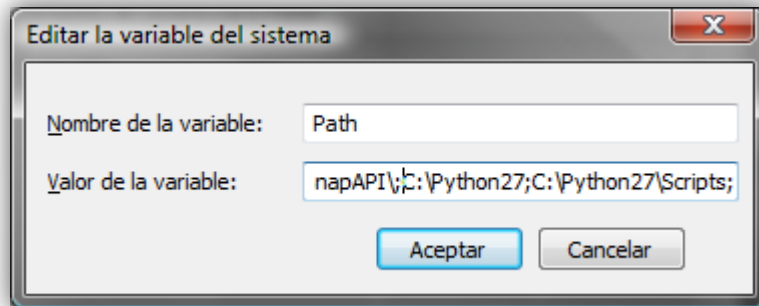


Ilustración 35 – Variable de entorno “Path”

- 6.8. Aceptamos en todas las ventanas abiertas y ya tendremos configurado Python para su uso.

2. INSTALACIÓN Y CONFIGURACIÓN DEL ENTORNO DE DESARROLLO: ECLIPSE + PYDEV

El entorno de desarrollo utilizado durante el desarrollo de este proyecto está formado por la herramienta principal Eclipse y el *plugin* PyDev disponible para Eclipse.

Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de *plugins*. No trabaja con un lenguaje específico, sino que es un IDE genérico, y mediante los diferentes *plugins* nos permite desarrollar software en diferentes lenguajes. PyDev es el *plugin* que permite desarrollar proyectos basados lenguaje Python mediante la herramienta Eclipse.

Por lo tanto, para poder utilizar Python con estas herramientas debemos instalar tanto Eclipse como PyDev y configurarlos para que utilicen el intérprete de Python instalado en la computadora.

1. Instalación de Eclipse [Cuarental Gazapo, Carlos. (2015)].

- 1.1. Identificar la versión del sistema operativo instalada en el ordenador:

En las propiedades de nuestro sistema podemos encontrar la edición de Windows que estamos utilizando (Windows Vista, Windows 7, Windows 8, Linux...) y el tipo de sistema operativo que utilizamos (32 o 64 bits).

- 1.2. Descargar Eclipse de la página oficial.



Podemos descargar la versión genérica de Eclipse en la web <http://www.eclipse.org/downloads/packages/eclipse-ide-eclipse-committers-442/lunasr2>. En los enlaces de descarga debemos seleccionar aquel que se ajuste a las condiciones del sistema operativo. También se pueden descargar otros paquetes de Eclipse que incluyen algunos complementos o *plugins* desde la página <http://www.eclipse.org/downloads/>.

- 1.3. El archivo descargado está comprimido. Extraer los ficheros almacenados en el archivo comprimido constituye el propio proceso de instalación. Por lo tanto, descomprimir el paquete descargado en el directorio que prefiramos. Recomiendo el directorio "C:\Program Files" para guardarlo como cualquier otro programa y tener un fácil acceso a él.
- 1.4. Crear un directorio que actuará como espacio de trabajo de eclipse. Este tipo de directorios suelen llamarse "workspace". Aún así, podemos llamarlo y ubicarlo según nuestras preferencias.
- 1.5. Ejecutar eclipse mediante el ejecutable "Eclipse.exe" ubicado en el directorio de instalación.
- 1.6. Nada más inicializarse el programa, nos solicitará que introduzcamos la ruta del espacio de trabajo. Introducimos la ruta del directorio creado anteriormente.
- 1.7. Ya podemos empezar a trabajar con Eclipse.

2. Instalación PyDev.

La instalación de PyDev se lleva a cabo desde la propia herramienta de desarrollo Eclipse, la cual nos da facilidades para la instalación de este tipo de *plugins*.

- 2.1. Abrir el programa Eclipse
- 2.2. En el menú superior nos dirigimos al submenú "Help y dentro seleccionamos "Install New Software...".
- 2.3. En la ventana de instalación que nos aparece presionamos "Add...".

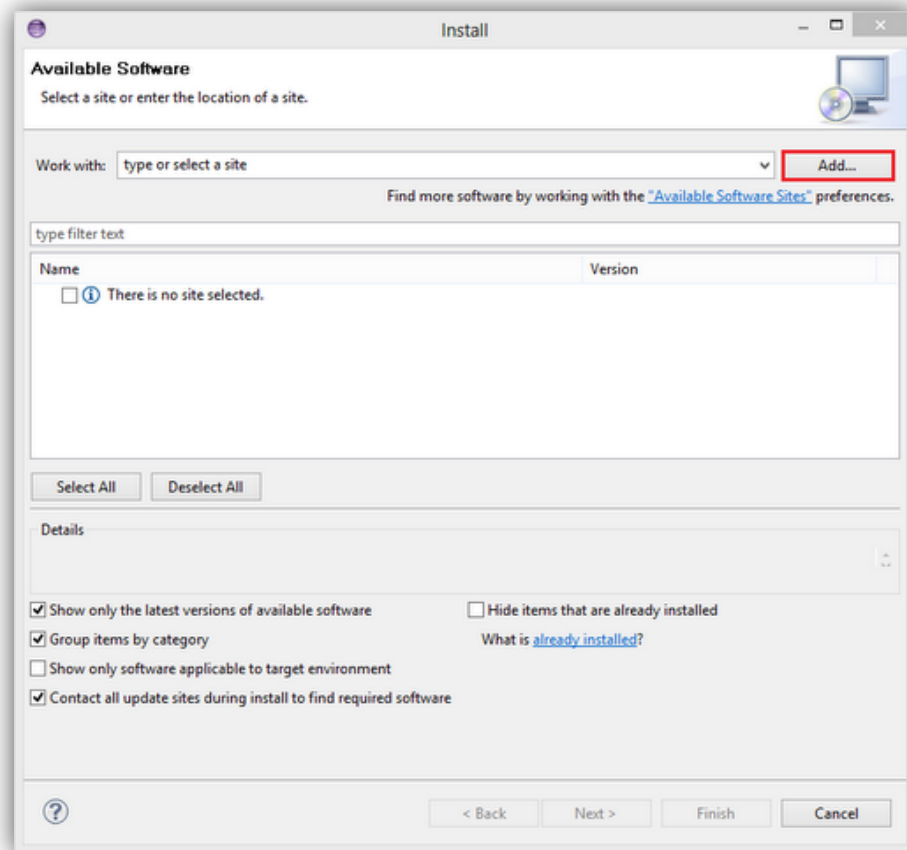


Ilustración 36 – Ventana de “Nuevo software” de Eclipse

- 2.4. En la ventana de añadir repositorio introducimos los siguientes datos de nombre y localización:

Name: PyDev and Pydev Extensions

Location: <http://pydev.org/updates>

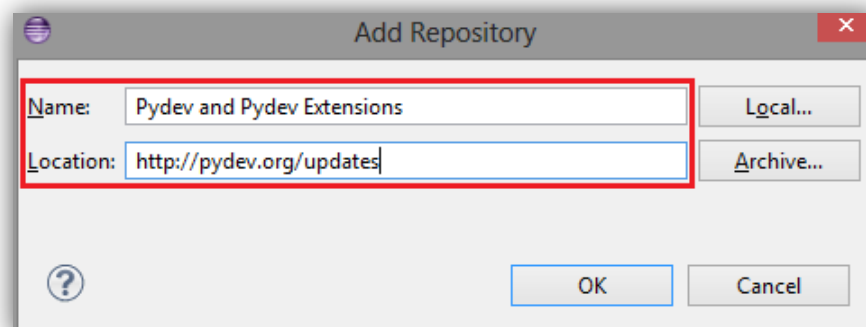


Ilustración 37 – Ventana “Añadir repositorio” de Eclipse

- 2.5. En la lista de paquetes para instalar que nos aparece seleccionamos sólo PyDev.

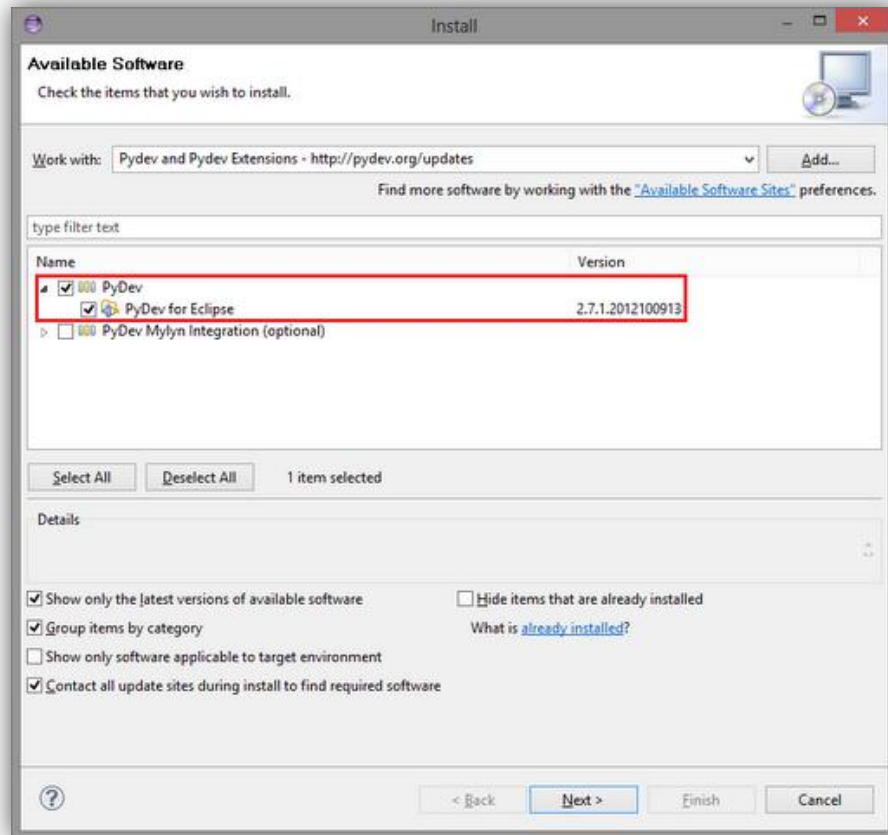
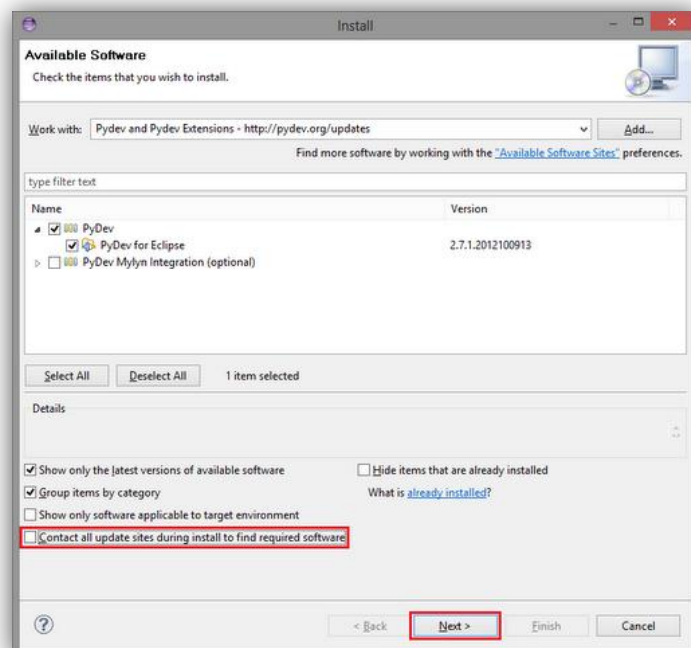


Ilustración 38 – Ventana de instalación de PyDev en Eclipse

2.6. Desmarcamos la casilla “Contact all update sites during install to find required software” y presionamos el botón “Next”.

Ilustración 39 -
Ventana de instalación
de PyDev en Eclipse (2)





- 2.7. Nos aparecerá una nueva ventana con los detalles de la instalación, presionamos “Next”.

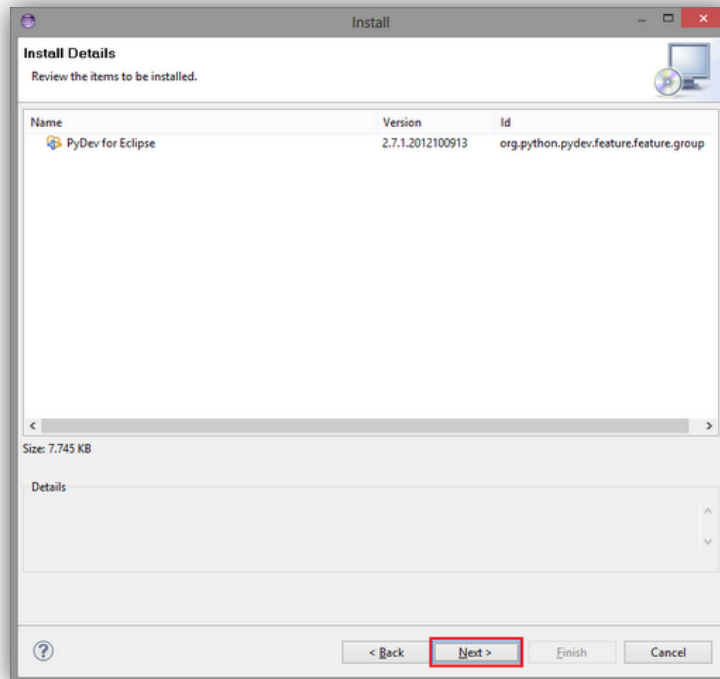


Ilustración 40 – Ventana de instalación de PyDev en Eclipse (3)

- 2.8. Aparecerá otra ventana con las condiciones de la licencia, seleccionamos la opción “I accept the terms of the license agreements” y presionamos “Finish”.

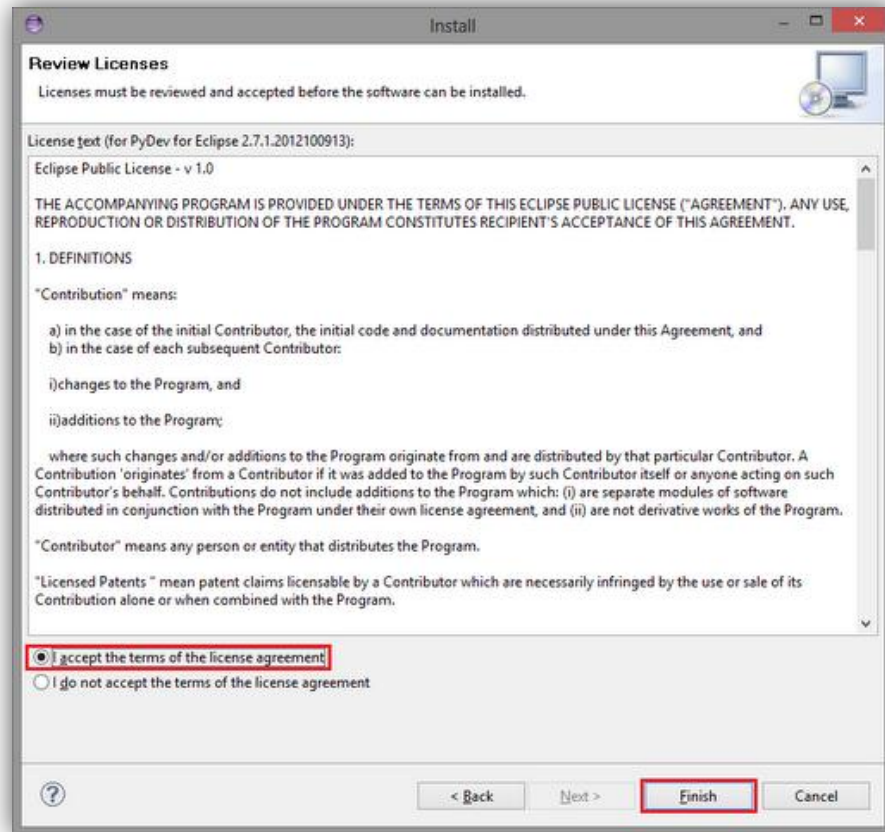


Ilustración 41 – Ventana de instalación de PyDev en Eclipse (3)

2.9. Esperamos a que acabe el proceso de instalación.

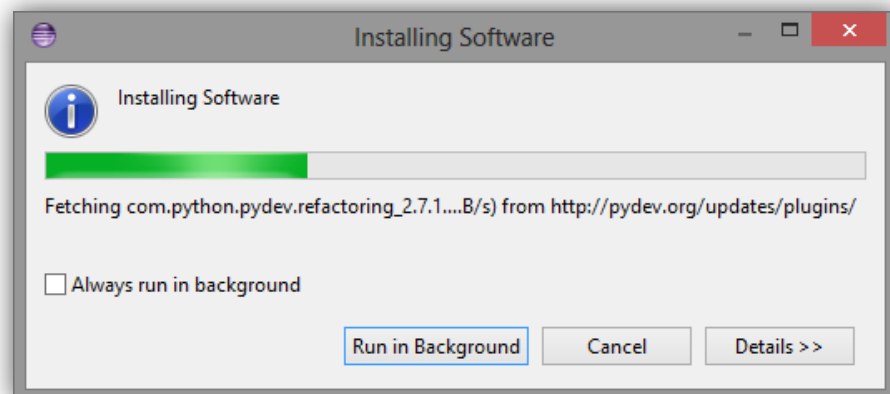
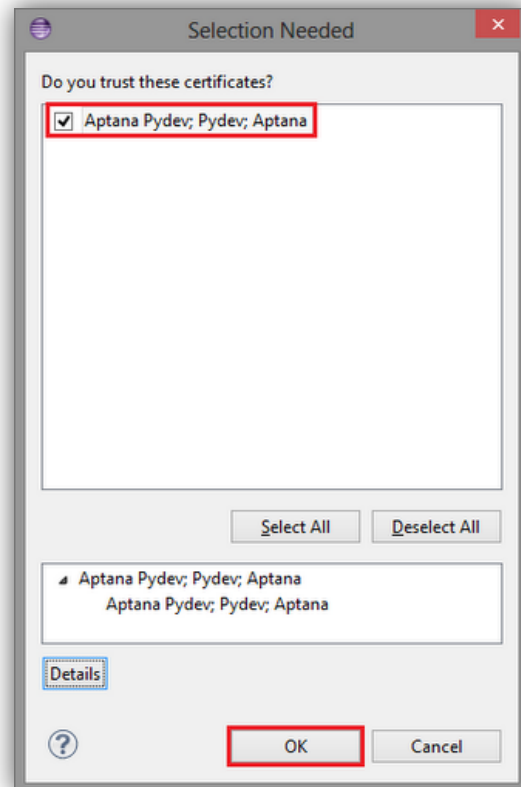


Ilustración 42 – Ventana de instalación de PyDev en Eclipse (4)

2.10. Si en algún momento nos pregunta si confiamos en el certificado de Aptana PyDev lo seleccionamos y confirmamos.



Ilustración 43 – Ventana de instalación de PyDev en Eclipse (5)



2.11. Por último reiniciaremos Eclipse, seleccionaremos “Yes” en la siguiente ventana.

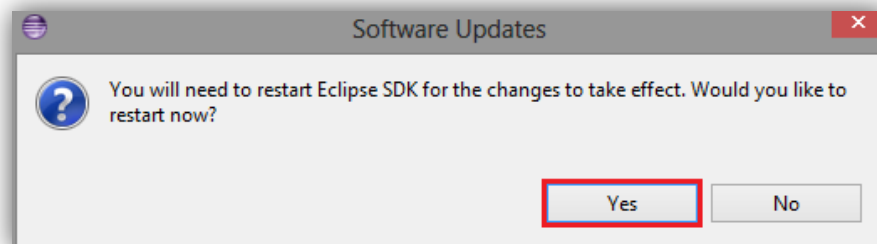


Ilustración 44 – Ventana de instalación de PyDev en Eclipse (6)

3. Configuración del intérprete

Una vez instalado el plugin PyDev debemos configurar el intérprete de Python en Eclipse. Para ello debemos realizar los siguientes pasos:

- 3.1. En el menú superior nos dirigimos al submenú “Window”. Dentro seleccionamos “Preferences”
- 3.2. En el índice de la ventana “Preferences” abrimos “PyDev” y seleccionamos “Interpreters” , y dentro “Python interpreter”

3.3. No centramos en la parte superior de la ventana, dedicada al intérprete. Ahora tenemos dos opciones:

- a) Seleccionar “Auto Config” (nos configura automáticamente el PATH de Python)
- b) Seleccionar “New” (para configurar manualmente el PATH de Python). Introducimos la ruta del fichero “python.exe “de la versión de Python instalada. (Recordemos que por ejemplo para la versión 2.7 de Python la ruta por defecto es “C:\Python27”)

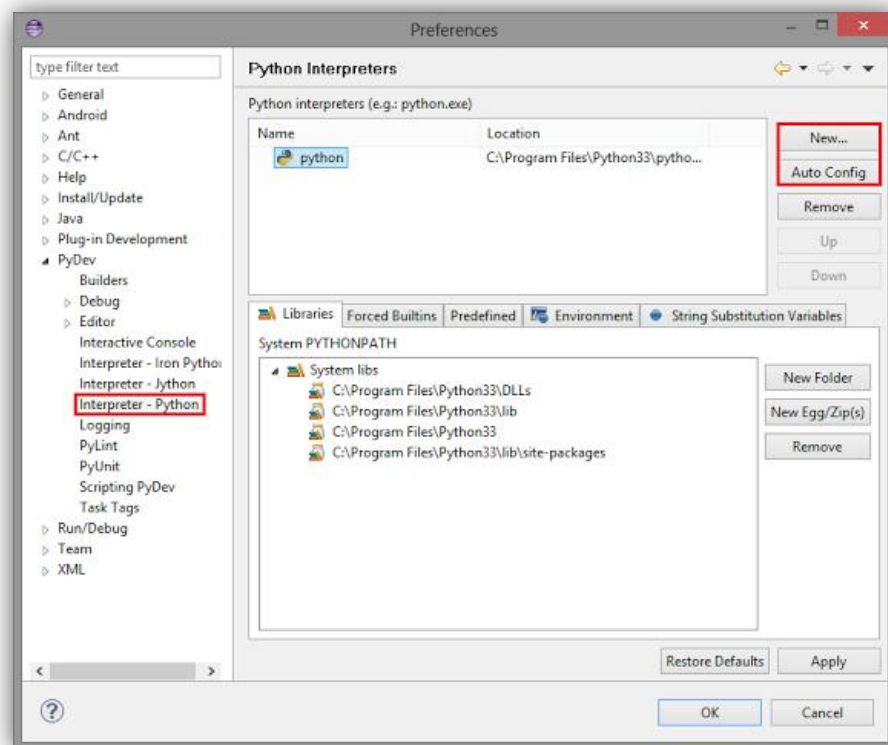


Ilustración 45 – Ventana de preferencias de Eclipse

- 3.4. Presionamos el botón “OK” para confirmar los cambios.
- 3.5. Con esto Eclipse está completamente configurado para utilizar el lenguaje Python mediante el plugin PyDev y el intérprete de Python.



3. INCORPORACIÓN DE LIBRERÍAS EXTERNAS

Para incorporar librerías externas al lenguaje de programación Python debemos descargarlas de sus respectivas páginas oficiales. Suelen venir con un instalador sencillo o en un paquete comprimido. Mediante estas utilidades debemos introducir los ficheros de la librería en la carpeta “site-packages” dentro de la carpeta “Lib” que está ubicada en la ruta de instalación utilizada para Python. De esta forma para una versión 2.7 de Python instalada en la ruta por defecto la ruta completa que del directorio que contiene las librerías de Python sería: “C:\Python27\Lib\site-packages”.

Una vez esté configurado Eclipse + PyDev para trabajar con Python y los archivos de las librerías externas en la ubicación indicada, podemos añadir librerías externas al entorno de desarrollo. Debemos indicar al intérprete de Python cuáles son estas librerías y dónde están ubicadas. Para ello el proceso es similar a la configuración del intérprete de Python del apartado anterior:

1. En el menú superior nos dirigimos al submenú “Window”. Dentro seleccionamos “Preferences”
2. En el índice de la ventana “Preferences” abrimos “PyDev” y seleccionamos “Interpreters” , y dentro “Python interpreter”
3. Nos centramos en la parte inferior de la ventana, dedicada a las librerías que puede utilizar el intérprete de Python. Presionamos sobre el botón “New Folder”.
4. En la nueva ventana seleccionamos la ruta donde esté ubicada la librería que pretendemos añadir. Es decir, la carpeta de la librería ubicada en el directorio “site-packages”. Presionamos “Aceptar” para añadir la ruta. Podemos añadir tantas librerías como necesitemos.

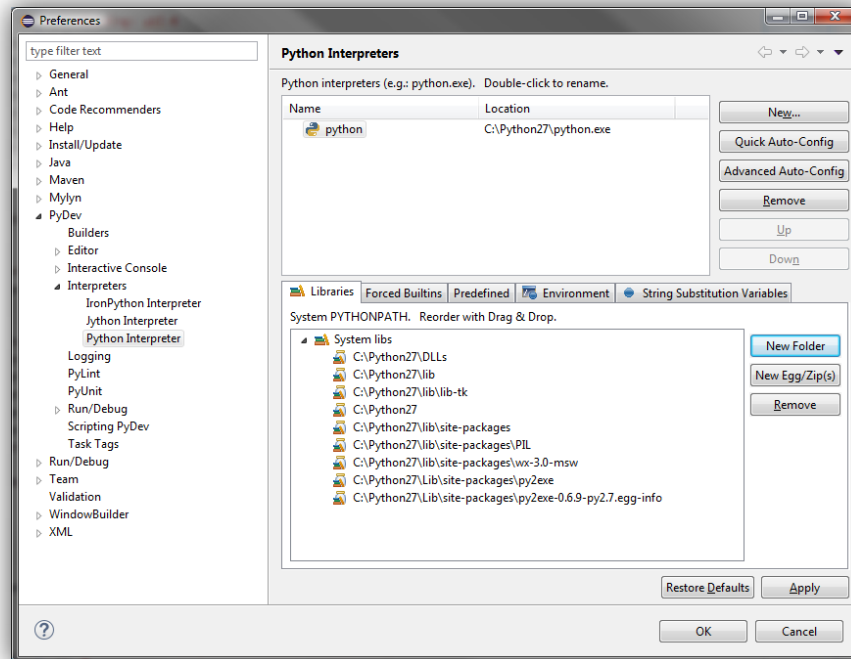


Ilustración 46 – Agregar librerías en Eclipse

5. Presionamos “Apply” para confirmar los cambios y “OK” para cerrar.
6. Ya podemos empezar a utilizar las funcionalidades incorporadas por las librerías externas.

4. CONFIGURACIÓN DEL SERVIDOR REMOTO

Durante el desarrollo de aplicaciones software se suele trabajar con servidores de forma local. Es decir, el servidor se encuentra activo en la misma máquina en la que se está desarrollando o probando el software. Pero en el caso de querer distribuir una aplicación o simplemente querer utilizar o probar la aplicación desde otra máquina diferente a la que contiene el servidor, debemos conectarnos al servidor de forma remota. Es decir, conseguir una conexión aplicación-servidor a través de internet estando estos ubicados en máquinas diferentes.

Como ya se ha mencionado anteriormente, para esta aplicación se utiliza XAMPP para proveer el servidor. A continuación se detallan los pasos [Carrodegua, N. (2015)] para configurar el servidor de forma remota mediante XAMPP y suponiendo que el ordenador que lo contiene está conectado a Internet mediante un único router. Se configurarán tanto el servidor “Apache” para proveer los servicios web como el servidor “MySQL” para ofrecer los servicios de conexión con la base de datos.

1. Comprobar que los servidores se encuentran activos y los puertos que están utilizando.

Para ello abrimos el panel de control de XAMPP e iniciamos los servicios si no están activos presionando sobre el botón “Start” de cada uno de ellos. Comprobamos que Apache y MySQL están utilizando los puertos TCP/IP establecidos por defecto, 80 y 3306 respectivamente.

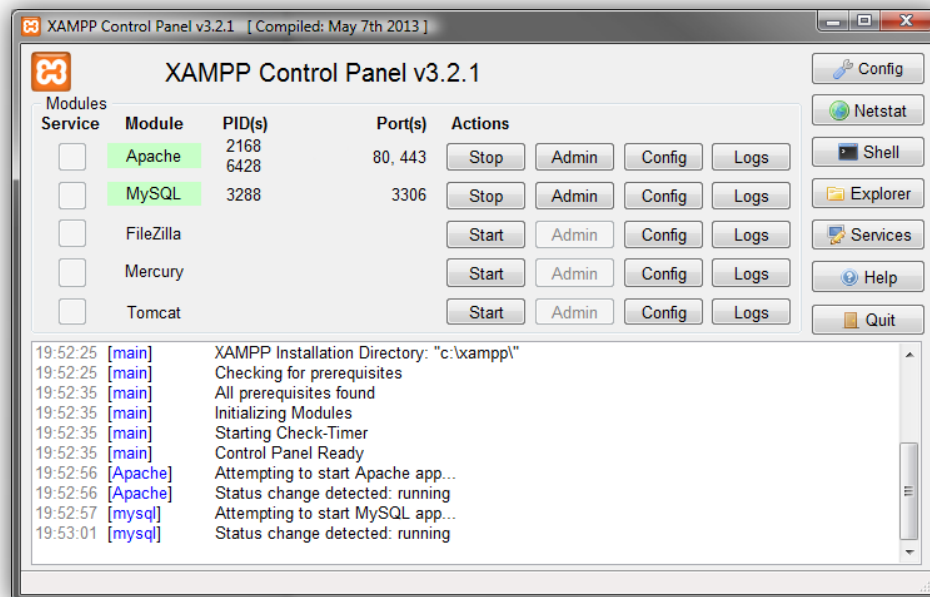


Ilustración 47 - Panel de control XAMPP con Apache y MySQL

2. Configurar los servicios Apache y MySQL mediante sus archivos de configuración.

2.4. Para configurar el servidor web Apache debemos editar el archivo de texto que guarda su configuración. Este archivo está nombrado como “httpd.conf” y se encuentra en la ruta “xampp\Apache\conf” partiendo de la ubicación donde está instalado nuestro servidor XAMPP. Lo podemos editar mediante cualquier editor de texto, aunque Notepad++ es el más recomendable. Tenemos dos opciones para editar este archivo:

a) Primera opción, la más sencilla, descarga una copia del archivo “httpd.conf” desde la web <http://norfipc.com/ftp/varios/httpd.zip>. Descomprímelo, cópialo o muévelo a la carpeta “xampp\Apache\conf” y



ANEXOS

sustituye el archivo original, ya tendrás listo para funcionar el servidor.

b) La otra opción, más avanzada pero no difícil, abre el archivo “httpd.conf” y edita manualmente las líneas que se indican:

i) La línea 52 Listen indica el puerto y dirección IP por el que el servidor va a recibir las peticiones. Sustituye “Listen localhost:80” (usado para recibir peticiones solo desde el mismo ordenador) por “Listen 80” para permitir la recepción de peticiones desde otras máquinas.

ii) En la línea 149 “DocumentRoot” es necesario especificar la ruta de la carpeta local que contendrá las páginas y archivos a servir. Esta carpeta es por defecto “C:\xampp\htdocs” (dependerá del directorio de instalación de XAMPP).

iii) La línea 177 <Directory> establece los permisos necesarios al directorio anterior, quedaría: <Directory “C:\xampp\htdocs”> (dependerá del directorio de instalación de XAMPP).

2.5. El servidor MySQL se configura de forma similar mediante su archivo de configuración “my.ini” ubicado en la carpeta “\xampp\mysql\bin”. Lo editamos manualmente:

En la línea 45 se establecen las direcciones IP que tienen permiso de conexión. Sustituimos la dirección IP que aparece tras “bind-address” por la dirección IP “0.0.0.0” para permitir el acceso desde cualquier dirección.

3. Reiniciar los servicios.

Para que los cambios efectuados tengan efecto debemos reiniciar los servicios de Apache y MySQL. Podemos cerrar el programa XAMPP y volverlo a abrir, para así reiniciar el servidor completamente. O mediante el panel de control de XAMPP paramos los servicios mediante el botón “Stop” y volvemos a iniciarlos mediante “Start” en cada uno de ellos.



4. Configuración de la base de datos.

En el caso del servidor MySQL debemos configurar la base de datos para posibilitar también el acceso externo. Podemos hacerlo mediante la herramienta phpMyAdmin un navegador web. Podemos abrirlo presionando el botón “Admin” correspondiente a MySQL que aparece en el panel de control. Nos dirigimos a la pestaña de usuarios y agregamos un nuevo usuario con el nombre de usuario y contraseña que queramos. En “Servidor” debemos seleccionar “Cualquier servidor”, y además debemos otorgar los privilegios necesarios sobre las bases de datos que queramos.

5. Configuración del router.

En el router que proporciona conexión a la máquina que aloja el servidor debemos llevar a cabo las siguientes configuraciones:

- 1.1. Establecer una dirección IP fija para la máquina que contiene el servidor. Será utilizada más adelante.
- 1.2. Abrir los puertos utilizados por Apache y MySQL (80 y 3306, respectivamente).
- 1.3. Elegir un puerto no utilizado para ningún servicio por defecto (uno para cada servicio que queramos activar). Por ejemplo, 4040 y 4041, que serán utilizados en la explicación de aquí en adelante.
- 1.4. Buscar la dirección IP pública del router.
- 1.5. Redireccionar los puertos de la siguiente forma:
IP pública, puerto 4040 → IP servidor, 80 (Apache)
IP pública, puerto 4041 → IP servidor, 3306 (MySQL)
- 1.6. Reiniciar el router para que los cambios tengan efecto

Estas configuraciones se llevaran a cabo de forma distinta en función del router. En mi caso, tuve que realizar la configuración en dos router diferentes, ya que un primer router ofrece conexión a un segundo, y es este el que está conectado con la máquina del servidor. La configuración de puertos en los routers se realizó de la siguiente forma:

Primer router:

- IP pública, puerto 4040 → IP segundo router, 4040 (Apache)
- IP pública, puerto 4041 → IP segundo router, 4141 (MySQL)

Segundo router:

- IP primer router, puerto 4040 → IP servidor, 80 (Apache)
- IP primer router, puerto 4041 → IP servidor, 3306 (MySQL)

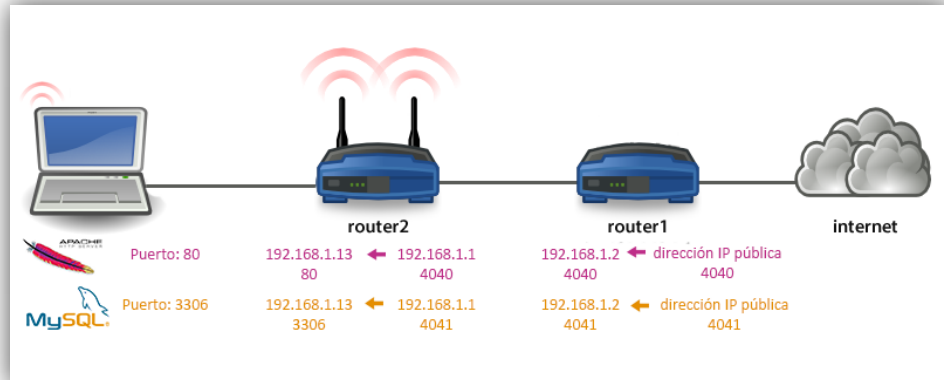


Ilustración 48 – Esquema de redirección de puertos

De esta forma, se crean “túneles” para que las peticiones de los servicios que lleguen al primer router se dirijan al servidor atravesando el segundo router, mediante la configuración mostrada en la *Ilustración 48*.

6. Conexión.

Finalmente, para conectarnos a ambos servicios debemos indicar la dirección IP y puerto:

- Para conectarlos vía web mediante un navegador de Internet al servidor web de Apache debemos introducir en la barra de direcciones:

“[dirección IP pública del router]:4040”

- Quando queramos realizar una conexión con el servidor de bases de datos MySQL debemos indicar:

Host: [dirección IP pública del router]

Puerto: 4041



Universidad de Valladolid

Diseño y desarrollo de una aplicación de escritorio dedicada a la composición fotográfica: Autodesign