

PORTARRETRATO DIGITAL

PUNTO C



ÍNDICE

1. Antecedentes de la propuesta.
2. Descripción del producto software.
3. Objetivos general y específicos del sistema.
4. Generar un diagrama de casos de uso, para representar la funcionalidad general del sistema.
5. Hacer una distinción clara entre los diferentes actores del sistema. (descripción de tipos de usuarios).
6. Elaborar las principales interfaces de usuario.
7. Describir, en forma clara, los principales requerimientos funcionales y no funcionales del sistema.

1-ANTECEDENTES DEL SISTEMA

La fotografía es un arte que muchos de nosotros disfrutamos plenamente, y que desde tiempos remotos hemos querido expresar y mostrar al mundo entero por medio de portarretratos que colgamos en nuestras paredes para recordar momentos especiales o a personas que tal vez no están cerca de nosotros.

Existen muchos marcos que nos permiten colocar estas imágenes; de madera, plástico y muchos otros materiales que nos permiten colocar nuestras más amadas fotografías, pero como todo evoluciona, la tecnología dio un gran paso, ¿Qué pasaría si en lugar de comprar muchos portarretratos para varias fotos, pudieses comprar uno solo y poner las fotos que quieras?, bajo esta premisa nació "El portarretrato digital".

1.1- Cambios en la sociedad

A comienzos de la década pasada, el mercado de la fotografía digital explotó y tardó muy poco en expandirse. Este fenómeno tuvo enormes efectos en las costumbres que hasta el momento el público había tenido con respecto a la fotografía analógica tradicional, y particularmente con las capturas impresas.

1.2 Tecnologías utilizadas

Compartir las imágenes se volvió mucho más barato e "instantáneo" con las nuevas conexiones ADSL, además de que el almacenamiento se abarataba a buen ritmo, convirtiéndose en un factor clave para alojar todas las fotos que se iban haciendo con las nuevas cámaras. Aunque frente al revelado y la impresión de fotos de forma amateur se sigue viendo como una opción, finalmente lo que triunfó y continúa vivo a día de hoy es: "guardar las fotos en el disco duro para ver más adelante".

1.3 Primeras marcas en desarrollar...

A partir de aproximadamente 2005, el mercado comenzó a expandirse, y grandes marcas del mundo de la imagen como Kodak, Philips, Samsung o Sony entraron al mercado junto con otros actores como Parrot, Ceiva o Nuvue. Más allá de que hubiera mercado, la comercialización masiva de los marcos digitales fue posible gracias a que la memoria flash que se hizo ubicua y sobre todo, a que los paneles LCD fueron bajando de precio con una mayor producción a medida que los viejos televisores CRT dejaban paso a la "gran" pantalla plana.

1.4 ¿Con qué se realizó?

La composición general de un marco de fotos digital era simple: una pantalla LCD de baja resolución para nuestros estándares actuales (800 x 480 píxeles como base en los modelos decentes), con un tamaño medio de entre 7 y 8 pulgadas, pero que partía desde las 3 hasta las 10 pulgadas. A ella se sumaba una memoria interna en la que almacenar el contenido multimedia, de entre 16 y 32 MB en 2007, un lector de tarjetas y/o entrada USB para copiar dicho contenido y un

software de control sencillo que permitiera la transferencia de archivos y configurar los modos de visualización de estos.

A esta base podían sumarse elementos de gama más alta, como conectividad inalámbrica Wi-Fi o Bluetooth para conectar a ordenadores y teléfonos móviles, salida de imagen para TV o monitor, alimentación con batería junto a toma de corriente y mando a distancia.

Eso en lo referente al apartado técnico. Por el lado estético, la oferta era más amplia. Las marcas podían jugar con marcos más grandes, con otros más reducidos, o con el estilo: líneas más tradicionales o más modernas. Los había más parecidos a tablets y otros que copiaban exactamente el estilo de los marcos de fotos tradicionales, integrando, eso sí, una pantalla en lugar de una imagen inerte.

2. Descripción del producto software.

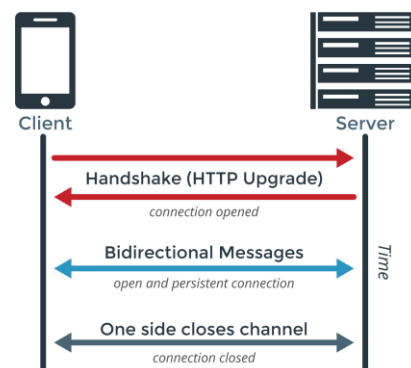
Usando nuestros conocimientos en lenguaje C, además de interesantes dispositivos de hardware, a continuación se describirá que se utilizará para realizar nuestro portarretrato digital, que a diferencia de los ya a la venta, tendrá una conexión directa con el usuario. El usuario utilizando una app, que fungirá como servidor podrá tomarse una foto y enviarla directamente al portarretrato, esté en el lugar en el que esté. Además de esta función en tiempo real, también podrá elegir alguna carpeta de su dispositivo móvil, para imágenes más personalizadas, El día de tú boda, tú graduación e incluso de tu mascota.

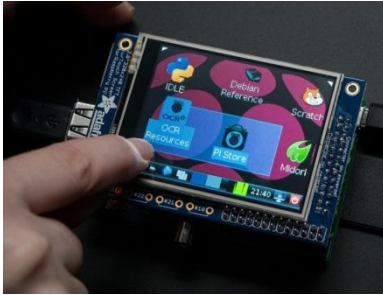
Uso de WebSocket

Tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP.

Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor.

Brindará la conexión servidor-portarretrato





Este display de 800x400 se conecta con la Pi usando tan solo dos conexiones, alimentacion desde un pin GPIO y un cable ribbon que se conecta al puerto DSI de la Raspberry Pi. Los drivers para el soporte tactil y el teclado integrado en pantalla se incluyen en la última version de Raspbian. Permite atornillarse cómodamente una Raspberry Pi sobre la controladora y queda un montaje muy compacto y profesional.

Mostrará las imágenes que el servidor le brinde*

Lenguaje C

Lenguaje de programación estructurada para la realización del cliente web socket y un poco de Processing 3.0 para el despliegue de la fotografía.

```
1 /* This line basically imports the "stdio" header file, part of
2  * the standard library. It provides input and output functionality
3  * to the program.
4  */
5 #include <stdio.h>
6
7 /*
8  * Function (method) definition. This outputs "Hello, world" to
9  * standard output.
10 */
11 void sayHello() {
12     // printf() is used to print the specified text (with optional
13     // formatting options) to the standard output.
14     printf("Hello, world!\n");
15 }
16
17 /*
18  * This is a "main function". The compiled program will run the code
19  * defined here.
```

3. Objetivos generales y específicos

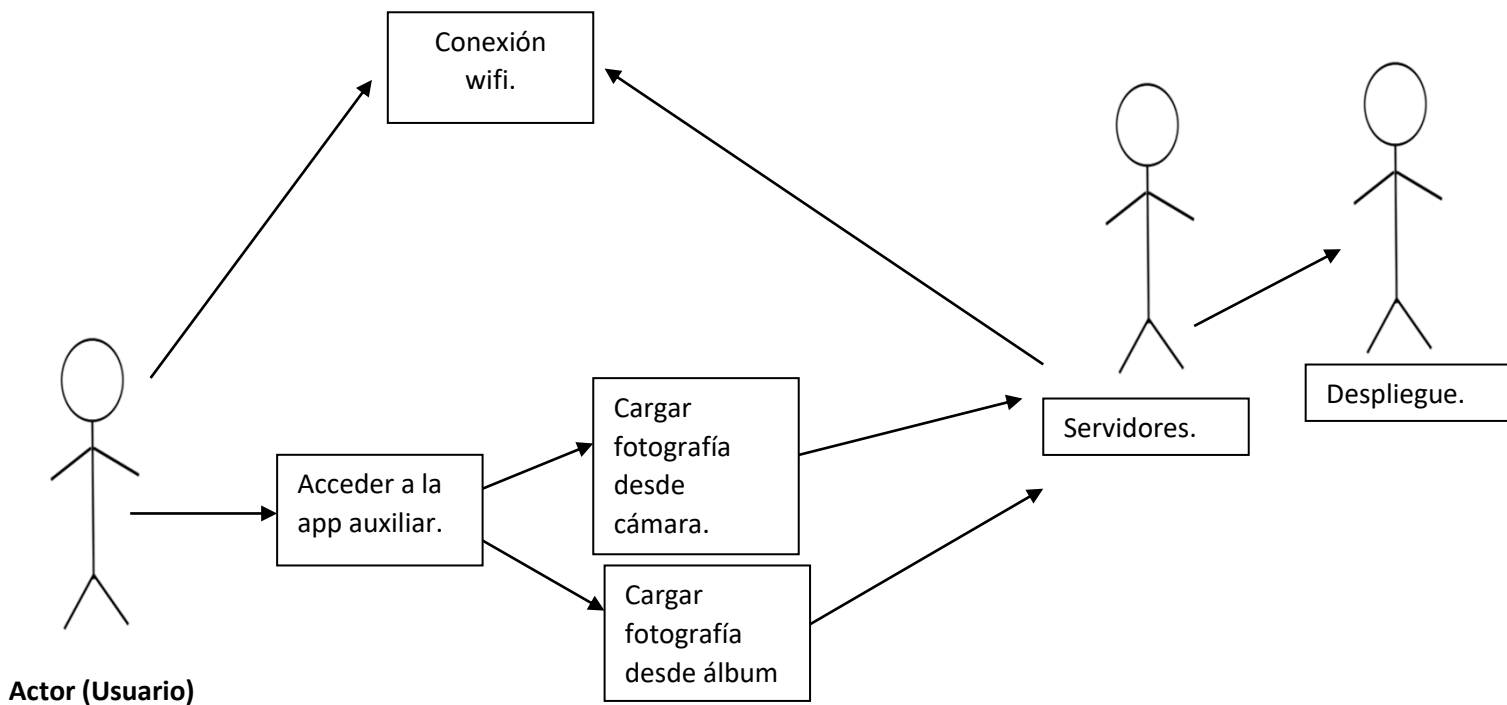
Objetivo General

El objetivo general del proyecto es desarrollar una aplicación que permita a los usuarios de gama media presentar sus fotografías de una manera distinta a la tradicional, en la cual usando la tecnología se deje a un lado la preocupación del maltrato o pérdida de fotos impresas para dar pie a un portarretrato digital que permita al usuario colocar infinidad de fotos sin tener que preocuparse por modificarla manualmente.

Objetivos específicos.

- Mostrar en tiempo real lo que estás viviendo, en dónde y con quién estás a quienes se quedaron en casa.
 - Personalizar las fotos según el evento que quieras recordar.
 - Ayudar a las personas que padecen Alzheimer a recordar a los integrantes de su familia.
 - Disminuir el uso de fotos impresas y la pérdida total de nuestras fotografías.
-

4. Generar un diagrama de casos de uso, para representar la funcionalidad general del sistema



Casos de uso

- El usuario accede a la galería de su dispositivo móvil para seleccionar una fotografía que sea de su agrado para desplegar en el portarretrato, por medio de la aplicación dicha foto llega a la de datos, donde el portarretrato la buscará y detectará para mostrarla en pantalla.
- Desde la aplicación externa el usuario accede a la cámara o a su galería de imágenes, si se elige la opción de acceder a la cámara, se deberá tomar una fotografía a la que se le debe de conceder autorización por parte del usuario para mandarla al servidor, mientras que si se selecciona la galería bastará con seleccionar una imagen o fotografía para que la aplicación la suba a la base de datos.
- El código estará contenido en el portarretrato (rapsberri) y solo será posible acceder a este por medio del dispositivo, el usuario solo necesitará instalar la aplicación externa y vincularla con el servidor. Todo el proceso se realizará por medio de conexión Wi-Fi. El usuario solo tendrá que seleccionar un archivo de imagen o tomar una fotografía, el proceso para que llegue dicho archivo al portarretrato será independiente al usuario. Una vez que el portarretrato cuente con las imágenes, se podrán realizar las acciones permitidas por cada perfil.
- El usuario podrá acceder al portarretrato por medio de dos diferentes perfiles que vendrán prediseñados en el código, por medio del perfil “administrador” podrá editar los parámetros con los que el portarretrato muestra las fotografías, además podrá editar la información de las fotografías e imágenes, así como su ubicación y las subdivisiones (álbumes) a las que pertenecerán. También visualizará otro perfil donde podrá seleccionar las imágenes a mostrar y el modo en el que se hará, pero no podrá editar directamente los ajustes del portarretrato ni la información de las imágenes, ambos perfiles serán capaces de acceder a las opciones que permiten agregar fotografías y mostrarlas en pantalla.
- Si el usuario lo desea, podrá agregar una reseña en cada una de las fotos o álbumes, dentro de la opción “galería” el usuario podrá encontrar la opción *editar* cuando seleccione una foto, selecciona un álbum o cree uno. En este apartado se mostrarán diversos campos como: Cambiar ubicación de la imagen (quitarla del álbum o pasarlo a otro), cambiar título, agregar reseña, etc.

Mapeo de Requerimientos

1. Descripción exhaustiva de los requerimientos funcionales y no funcionales del sistema.

Requerimientos Funcionales

RF_01: El programa debe dar la opción de crear y modificar álbumes donde se almacenen fotos que el usuario designe, de tal modo que se pueda seleccionar alguno de estos y que las fotos contenidas se muestren en el portarretratos. Las fotografías deben de poder seleccionarse desde una interfaz que contenga todas las fotos no pertenecientes a un álbum o desde el contenido de los álbumes.

RF_02: Cuando se seleccione una foto o un álbum el programa debe desplegar una serie de opciones, donde el usuario podrá establecer parámetros que le servirán al programa al momento de mostrar la foto, opciones como: Tiempo que se mostrará la foto, intervalos de tiempo para cambiar las fotográficas, pausar en una sola imagen, comenzar o detener un modo aleatorio para las fotos, etc. *{Segunda versión, aún no implementado}*

RF_03: Se le permitirá al usuario la opción de agregar una fotografía desde la cámara o desde un álbum, accediendo desde una app auxiliar (moments), de tal modo que pueda editar la fotografía desplegada en el display.

RF_04: Siempre debe existir una opción en pantalla que despliegue un menú con diferentes opciones dependiendo de donde se encuentre el usuario, algunas de estas son: Detener (en caso de modo aleatorio), “cambiar la fotografía” en el caso de que se quiera intercambiar la imagen mostrada por una contenida en el dispositivo, editar información de la foto actual o salir (llevar al *home*).

RF_05: Debe ser posible visualizar una lista de la fotográficas y álbumes existentes como si se tratara de una galería, esta opción se debe de mostrar al ejecutar el programa después de haber iniciado según el usuario que se desee, llevando al usuario a otra interfaz (interfaz #2) donde podrá ver los archivos de fotos que existan en el dispositivo, seleccionarlos y al hacerlo se mostraran opciones de edición y la opción de mostrarla en pantalla.

RF_06: Cuando no se quiera utilizar alguna de las fotografías en el portarretrato, el usuario podrá encontrar una opción que le permitirá cargar una fotografía para mostrar en pantalla. Dicha opción podrá utilizarse en cualquier momento ya que siempre aparecerá en pantalla o en algún menú. Esta opción realizará la misma acción, pero de diferentes formas, ya que si se selecciona en cualquier interfaz el programa buscará nuevas actualizaciones de las imágenes contenidas para mostrarlas en la galería, mientras que en la opción que permita esperar una fotografía, esta se mostrará automáticamente cuando se detecte una actualización. {Usando la app auxiliar se podrá interrumpir o cambiar la fotografía}

RF_07: Al ejecutar el programa se visualizará una interfaz (interfaz #1) conocida como *home*, donde el usuario tendrá la opción de ingresar a la galería donde podrá seleccionar fotos a mostrar, cargar nuevas fotografías, editar archivos, establecer los parámetros de imagen cuando se seleccione la opción de “mostrar en pantalla”, etc. Se trata de la primera interfaz que el usuario podrá visualizar inmediatamente después de haberse validado el perfil con el que se entrará al programa. {App auxiliar}

RF_08: Debe existir una opción llamada en *tiempo real* o *esperar foto para visualizar*, donde el programa esperará a que el usuario mande una foto al portarretrato desde la aplicación del dispositivo móvil del usuario, el portarretrato conectado a internet recibirá la fotografía y este la mostrará de inmediato. Este modo solo mostrará una fase de “espera” cuando se seleccione, ya que después bastará con que el usuario mande otra foto para que la foto se intercambie.

RF_10: Será necesario utilizar una aplicación diferente al programa a entregar, esto con la finalidad de definir de mejor forma las características, acciones y objetivos del código, dicha aplicación servirá para ser un intermediario entre la base de datos donde se almacenarán las fotografías y el código del proyecto, el dispositivo descargará la fotografía en el sistema para que el código pueda detectar las imágenes contenidas y mostrarlas.

RF_11: Es necesario estandarizar un formato de imagen para los archivos que se manipularán, ya que programar el código de tal forma que acepte cualquier tipo de archivo de imagen podría generar problemas que dejarían a un lado el propósito

del programa y los objetivos planteados, el programa debe ser capaz de detectar el formato (o los formatos, dependiendo del proceso y las decisiones durante la codificación) de las imágenes que estén contenidas en el dispositivo y realizar todas las acciones con de estas.

RF_12: Es necesario el uso de un sistema operativo más amigable y flexible para realizar la codificación del programa y lograr que las librerías funcionen, ya que, a lo largo de la investigación de antecedentes y experiencia de los miembros del equipo, se ha identificado que existen mejores herramientas de las planteadas originalmente para la realización del proyecto, se propone el uso de Linux ya que presenta una mejor compatibilidad con las librerías y herramientas necesarias.

RF_13: Dentro del proceso para llevar la imagen desde el dispositivo del teléfono hasta el portarretrato, será necesario utilizar dos servidores, uno de carga de imagen, que almacenará las fotos, y un servidor libwebsocket. El portarretrato utilizará conexión a internet para estar vinculado a la base de datos mientras el código se está ejecutando, las funciones del programa validarán las imágenes contenidas y determinarán las acciones que se realizan y detectarán nuevas imágenes en el dispositivo.

RF_14: Las interfaces deben de tener un diseño que apoye a las funciones que el código tendrá, permitiendo que las acciones que el usuario realice sean bastante intuitivas, por lo que el programa debe de ser lo suficientemente simple, pero a la vez tener elementos que guían al usuario o que permitan que este sepa lo que está haciendo, permitiendo interactuar con el código sin complicaciones.

RF_15: Para permitir que el usuario tenga mayor seguridad en los cambios que este realice en el código, se desplegará una interfaz especial al ejecutar el programa, dicha interfaz le dará dos opciones al usuario, una le permitirá realizar todas las acciones planteadas en los requerimientos, principalmente las que están relacionadas con la edición de la información de las fotografías y la carga de estas, así como de la edición de álbumes y la creación de estos. La segunda opción solo desplegará las opciones para visualización de imágenes y álbumes,

tener acceso a la información de estos y elegir los elementos a mostrar en el portarretrato.

RF_16:El usuario debe poder tener acceso a una opción que le permita entender tanto el objetivo del programa, como todas las funcionalidades con las que cuenta y lo que ofrece, desde esta opción podrá tener acceso a un tutorial, donde se encontrara la información planteada anteriormente, además de guías paso a paso de cómo utilizar el programa y dar a conocer las funcionalidades, también se debe contemplar que en este apartado se den a conocer las limitaciones del programa.

RF_17:La opción que desplegará el tutorial para comprender el uso del programa deberá estar segmentada en diversos subtemas que tratarán las distintas funcionalidades del programa, irán desde la más simple hasta la más completa. Los “temas” estarán divididos por funcionalidades del código, al seleccionarlas aparecerá la información de dicha funcionalidad, así como la forma acceder a esta. Un subtema serán las funcionalidades, otro las restricciones del programa y otro contendrá información general del programa, se encontrará información sobre la finalidad del programa y sus características

RF_18:Las interfaces de usuario deben ser lo más intuitivas y amigables posibles, deben ser un apoyo para que el usuario explore las distintas funcionalidades del programa sin que este tenga complicaciones. Diversos elementos como un menú desplegable que cambiará según la funcionalidad en la que se encuentre, esto limitara lo que el usuario pueda realizar en ciertas partes, pero permitirá que este aprenda las opciones y restricciones de cada sección. La visualización de la ruta del código en la cual se está permitirá que el usuario siempre sepa en qué parte del se encuentra, además de otros elementos le permitan al usuario saber qué es lo que está haciendo.

RF_19:Con el fin de ayudar a la experiencia del usuario, dentro de los menús desplegables debe de haber una opción que permita dirigirse directamente al “home” independientemente de la parte del programa donde se encuentre, en el caso de que se encuentre en las opciones de manipulación de imágenes, se le indicara al usuario cuáles serán las consecuencias de ir al home sin haber guardado algún cambio.

RF_20: Como se ha planteado anteriormente, una de las opciones de manipulación de imágenes debe ser el poder agregar títulos o pequeñas reseñas de las fotografías, esta opción debe permitirle al usuario poder identificar las imágenes por medio de sus propios títulos. El programa debe reconocer cuando alguna imagen, titulo o álbum ya es encuentra bajo un nombre, de forma que no se creen varios archivos con el mismo nombre y genere conflicto en la ejecución de las funciones del código.

RF_21: Cuando se esté visualizando una imagen, el programa siempre debe tener una opción que permita salir de la imagen para regresar al programa tal y como estaba, esta opción debe encontrarse en cualquier modo de visualización de imágenes, ya sea visualizar una sola fotografía, un álbum o una selección aleatoria de fotografías.

RNF_22: El usuario debe de proporcionar una clave especial que le permita acceder al usuario de edición. De tal forma que la información que se tenga en el programa no sea modificada, también existirá la opción de cambiar la clave en caso de que el usuario así lo requiera.

RNF_23: Se podrá visualizar una imagen de pantalla de carga cuando se ejecute la opción de esperar foto, del tal modo que eso será lo que el usuario visualizará hasta que aparezca una nueva fotografía en la base de datos, al cambiar la fotografía esta pantalla de carga podrá aparecer dependiendo de la opción que el usuario haya seleccionado. Dicha pantalla de carga debe hacer referencia a algún elemento importante del proyecto.

RNF_24: En caso de que se presente algún error por parte del programa, alguna opción que el usuario haya seleccionado o parámetro que haya ingresado, además de tomar en cuenta fallas en la aplicación externa o por parte del servidor y la conexión, el código mostrará un mensaje de error que indicará que algo ha salido mal dentro del proceso.

Requerimientos No Funcionales

RNF_01: El programa externo utilizará la conexión Wi-Fi del dispositivo para subir las fotografías al servidor de carga, el dispositivo con el programa descargará las fotos y el código detectará la nueva imagen, de esta forma las fotografías se mostrarán en tiempo real, de tal modo que el usuario pueda tener conocimiento de lo que se está mostrando en el portar retrato sin necesidad de estar junto a este. Tanto el dispositivo móvil del usuario y el portarretrato necesitarán conexión a internet para realizar el proceso planteado en los diagramas.

RNF_02: El programa debe de validar que los títulos de las imágenes, tanto de las renombradas como de las que se el programa detectará, no se repitan, de tal forma que no exista conflicto entre las imágenes, esto se realizará mediante un arreglo que contenga los nombres de todas las imágenes. Dependiendo de la organización de las imágenes o de las decisiones de la codificación esta opción se realizará con vectores o matrices. *{Segunda versión, aún no implementado}*

RNF_03: Los álbumes deben contener una estructura que permita al usuario escribir reseñas generales sobre el contenido de este, además de poder asignarle un nombre al álbum. Se debe considerar que cada foto seleccionada será contenida en la estructura del álbum con su propia información contenida en la estructura de dicha fotografía. Se busca que cada fotografía sea única independientemente de que se encuentre contenida en un álbum. *{Segunda versión, aún no implementado}*

RNF_04: Se utilizarán estructuras y apuntadores para ubicar rápidamente los elementos que el usuario quiera, además de permitir la visualización de estos. Los álbumes serán organizados mediante arreglos de estructuras que organizarán los datos del portarretrato. Por medio de memoria dinámica se editará el tamaño de los arreglos que almacenen información específica de los elementos contenidos.

RNF_05: Para ayudar al usuario a encontrar las fotografías, se añadirá una opción que permita acceder a los arreglos del programa para encontrar un elemento en específico, esta búsqueda se podrá hacer por medio del título de la imagen, su fecha o algún otro dato, se imprimirán los datos de las fotografías (los más

simbólicos) y el lugar donde están contenidas. Esta opción también se podrá encontrar en los álbumes. *{Segunda versión, aún no implementado}*

RNF_06: El programa debe ser bastante simple de utilizar ya que no está pensado específicamente para 1 solo tipo de público, por lo que las interfaces de usuario, los nombres de las opciones y otros elementos no deben ser muy complicados. Además de que se implementarán herramientas que ayuden a la experiencia del usuario y lo guíen para aprender a utilizar todo lo que el programa ofrece.

RNF_07: Las funciones del código también representaran las diversas herramientas que este tenga, la validación de diversos datos importantes también se realizará por medio de funciones. Se definirán variables con nombres y se les asignarán funcionalidades específicas que intervendrán en funciones como: “esperar foto”, “editar información” etc.

RNF_08: En necesario organizar todo el código en varias funciones que cuenten con la característica de poder convocar a otras funciones, para que la interacción del usuario y el recorrido de este en el código se vea de forma clara mientras se esté ejecutando el código. Además, siempre debe de estar habilitada una opción que permita que el usuario acceda rápidamente el “home”, interrumpiendo lo que esté realizando.

RNF_09: Con el fin de que el programa sea lo más intuitivo posible y que este dotado de herramientas que apoyen el usuario, existirá un manual de usuario dentro del programa, esta opción se mostrará antes poder visualizar cualquier interfaz que no sea la principal, formara parte de las primeras opciones que se muestran, al ingresar se podrán encontrar guías e información importante del proyecto.

RNF_10: La opción de tutorial se encontrará dentro del menú principal del programa, por medio de arreglos se mostrarán los diversos tutoriales, es necesario clasificar cada una de las funcionalidades para posteriormente acomodarlas en diversos mini tutoriales, de tal forma que inicie en los más básico y termine en lo más complejo del programa. *{Segunda versión, aún no implementado}*

RNF_11: El programa debe reconocer que el archivo que se está queriendo mostrar en el portarretrato sea una fotografía, de no ser así se debe mostrar un

mensaje que advierta que hubo un error o el archivo no es el que se espera. Los parámetros que dependan de valores ingresados por el usuario deben contar con una función que valide el dato ingresado.

RNF_12: Se debe implementar un ícono que se mostrará siempre en la pantalla (por lo que no debe ser molesto), este elemento desplegará un menú que dependerá de la interfaz donde se encuentre el usuario, pero contará con las herramientas que se apeguen más a las necesidades que requiera el usuario en ese momento, cabe resaltar que la opción de ir al “home” siempre estará presente, independientemente de lo que se esté haciendo. *{Segunda versión, aún no implementado en el display}*

RNF_13: El programa debe tener una serie de pasos bien definidos para acceder a ciertas funciones de este, opciones como salir, editar, guardar, esperar foto, crear álbum, etc., deben contar con pasos adicionales que permitan identificar si lo que se está realizando el código es realmente lo que se quiere. Además de impedir una salida forzada del programa de lo haga fallar. Esto con la intención de que elementos importantes que el programa necesita guardar para conservar lo que el usuario ingrese no se pierda.

RNF_14: Será necesaria la generación de archivos de texto organizados, de tal modo que el programa pueda leerlos y utilizarlos para ejecutarse tanto con las opciones personalizadas, como con las descripciones tanto de fotos como de álbumes que el usuario haya anteriormente establecido. Estos son los datos de vital importancia que requieren de una serie de pasos bien definidos para realizarse, ya que es la configuración del usuario y se debe de priorizar su almacenamiento.

RNF_15: El cliente libwebsocket, deberá contener los protocolos de conexión a ambos servidores para que se comuniquen entre sí, utilizando conexión Wi-fi, permitiendo la carga, el aviso y el despliegue de la fotografía que el usuario proporcione.

RNF_16: El programa debe seguir ejecutándose sin problemas mientras se está mostrando la foto, la función del menú de opciones debe estar siempre activa, lista

para indicarle al programa cuales son las acciones que el usuario desea. Esta opción está pensada para que la imagen pueda mostrarse teniendo siempre un par de elementos en pantalla que lleven al usuario el “home” o que le permitan cambiar la fotografía.

RNF_17: Junto al menú desplegable se encontrará una ruta para guiar al usuario de tal forma que le permita saber lo que está realizando, lo que se realizará será “mapear” las funciones, de tal forma que el código permita imprimir las funciones anidadas que se están ejecutando en ese momento y que las imprima en un formato especial, con el fin de hacerlo parecer una ruta. Esto se realizará por medio de un printf especial que se imprimirá en una parte específica del código, mismo que cambiará según se navegue por el programa.

RNF_18: Se debe verificar que la aplicación externa pueda colocar debidamente las imágenes en el dispositivo de tal forma que el programa solo tenga que detectar las que se encuentran almacenadas en una parte específica del equipo. Para esto se modificará la ruta de descarga donde el portarretrato por medio de la base de datos descargará y archivará el elemento a mostrar en pantalla.

RNF_19: El programa solo debe depender de la aplicación externa para obtener los archivos, en lo demás, la aplicación debe de ser completamente independiente, se busca que el programa solo se cierre o ejecute una acción en concreto si el usuario así lo desea, sin permitir que los demás elementos que intervienen modifiquen o causen estragos en el código.

RNF_20: Para evitar complicaciones entre las aplicaciones, el programa se deberá de cerrar para poder realizar la carga de nuevas imágenes por medio de la aplicación externa, después el proceso se realizará tal cual se ha planteado anteriormente. Es importante recalcar que esta acción se realizara de esta forma solamente si la finalidad no es mostrar en pantalla (en ese momento) los archivos que se descarguen, ya que el proceso de que las imágenes lleguen al portarretrato es ajeno al código que las muestra en pantalla.

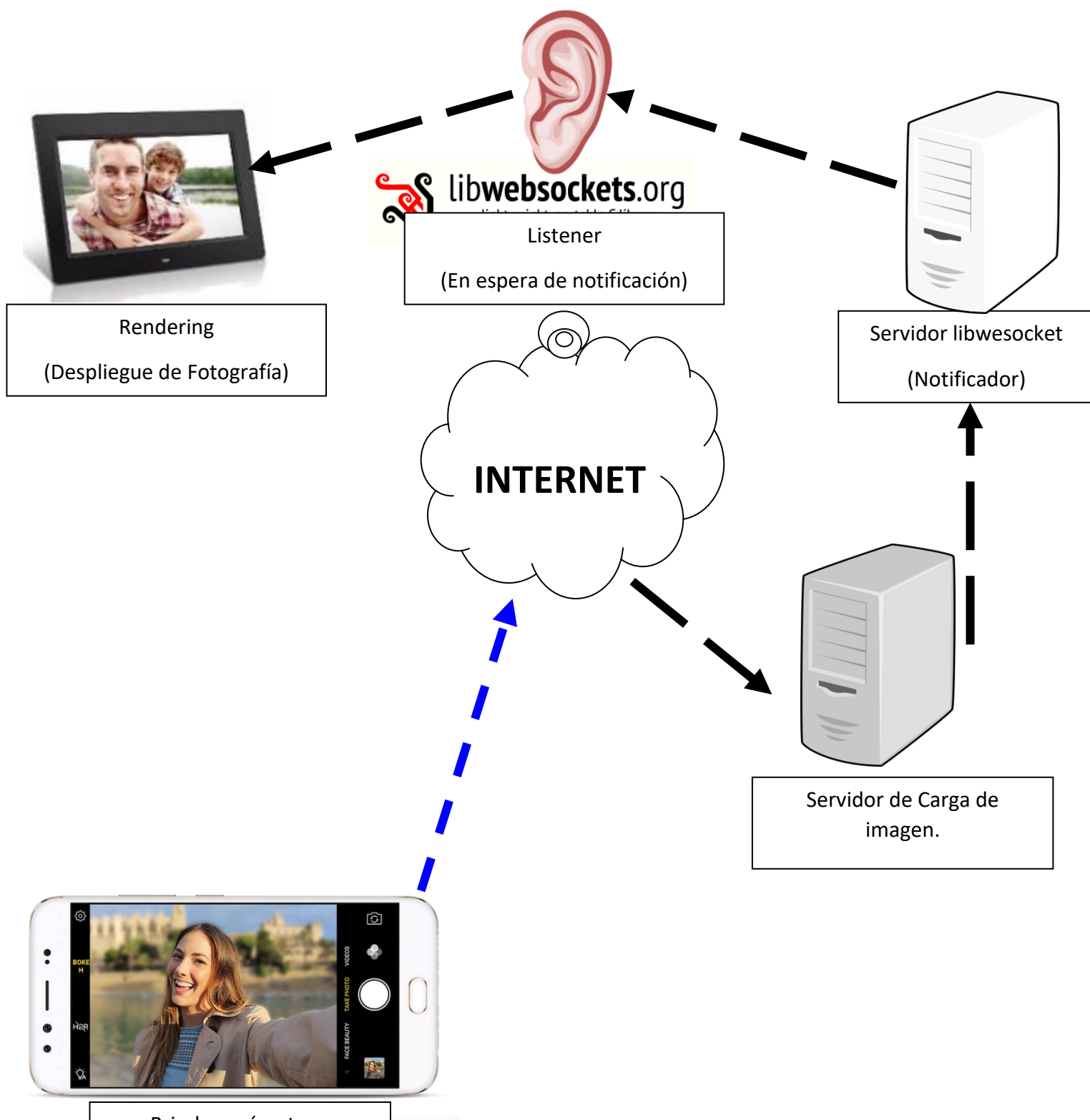
RNF_21: La sintaxis del código debe de hacer referencia a imágenes y no referirse siempre a los archivos como fotografías, ya que dadas las

especificaciones del código es posible seleccionar imágenes de la galería del dispositivo móvil del usuario, por lo que los elementos contenidos en el portarretrato o los mostrados en pantalla no necesariamente serán fotografías que el usuario haya realizado.

RNF_22: Uso de la librería “libwebsockets” para poder codificar acciones y funciones del cliente (Listener), que permitirá la conexión al servidor de carga de imagen y al servidor libwebsocket, además de que contiene las herramientas necesarias de manipulación y visualización de imágenes necesarias.

2. Elaborar los casos de uso para cada funcionalidad del sistema, donde se describa el proceso y las rutas de acción alternativas.

Estructura del sistema.



-Seleccione el Requerimiento de su preferencia, este lo dirigirá a su descripción-

Matriz de Requerimientos						
<i>Requisitos Funcionales</i>						
	Objetivo General	Objetivo Especifico 1	Objetivo Especifico 2	Objetivo Especifico 3	Objetivo Especifico 4	Prioridad
RF_01	★		★	★		2
RF_02	★	★	★	★		2
RF_03	★		★	★	★	2
RF_04						2
RF_05	★	★	★	★		2
RF_06	★	★		★		2
RF_07		★			★	2
RF_08	★	★			★	1
RF_09	★				★	1
RF_10		★			★	1
RF_11	★				★	1
RF_12	★	★	★		★	1
RF_13	★	★			★	1
RF_14		★		★		2
RF_15	★		★	★	★	2
RF_16	★			★		3
RF_17	★		★		★	2
RF_18	★			★	★	3
RF_19		★		★		2
RF_20	★		★	★		2
RF_21	★				★	1
RF_22	★				★	2
RF_23	★	★			★	1
RF_24	★		★		★	1
<i>Requisitos No Funcionales</i>						
	Objetivo General	Objetivo Especifico 1	Objetivo Especifico 2	Objetivo Especifico 3	Objetivo Especifico 4	
RNF_01			★		★	1
RNF_02		★	★	★	★	2
RNF_03	★		★	★	★	2
RNF_04	★	★	★			1
RNF_05		★		★		3
RNF_06	★			★	★	3
RNF_07					★	1
RNF_08					★	1
RNF_09	★	★		★	★	3
RNF_10	★		★			3

RNF 11	★			★	★	3
RNF 12		★			★	1
RNF 13	★			★	★	2
RNF 14	★				★	1
RNF 15	★	★	★			2
RNF 16				★	★	2
RNF 17		★		★	★	1
RNF 18	★			★	★	2
RNF 19	★				★	1
RNF 20		★		★		1
RNF 21	★	★			★	2
RNF 22	★			★	★	2
RNF 23		★			★	1

4-Informe del avance general de la implementación de dichos requerimientos.

Varios requerimientos han sido modificados ya que se reescribieron para que estén mejor relacionados con los objetivos del proyecto, también se agregaron nuevos requerimientos que se planteaban durante la codificación y planificación de las funcionalidades del programa. Se contaba con una cantidad mayor de requerimientos, pero una parte fue eliminada ya que eran redundantes o hacían referencia a funcionalidades que el programa podría tener a largo plazo, no eran referentes a las herramientas básicas con las que contaría el proyecto.

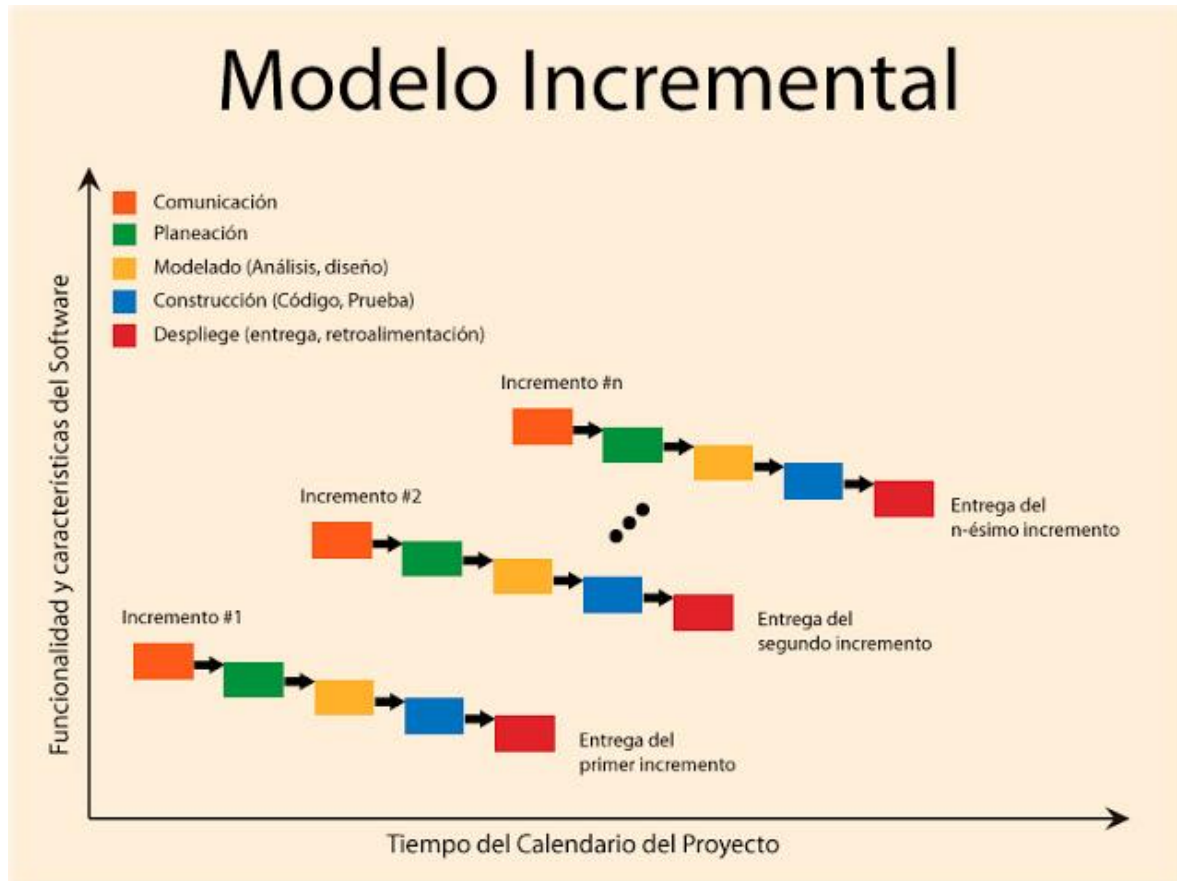
Los requerimientos no funcionales aumentaron debido a las modificaciones hacia los requerimientos funcionales, muchos fueron modificados, eliminados y editados ya que no encajaban con lo descrito en los casos de uso, los objetivos o algún requerimiento funcional. La matriz de requerimientos ayudo a editar la cantidad de requerimientos que se tenían, ya que es una herramienta visual que nos ayudó a darnos cuenta de lo que hacía falta y, sobre todo, de lo que sobraba en los requerimientos.

A diferencia de la primera entrega, ahora los requerimientos abarcan más temas relacionados con la experiencia del usuario, además de contar con definiciones claras y herramientas necesarias para la codificación. Se han colocado requerimientos que describen la experiencia del usuario a través de las funciones del programa, se plantearon soluciones y herramientas que ayudarán al programa a solucionar conflictos que sintaxis, errores en los archivos, entre otros.

Los requerimientos también reflejan los cambios que hemos tenido que realizar para poder llevar el proyecto hasta lo que es ahora, ya que se describen los sistemas operativos a usar, librerías y formas de organizar el código. Además, se plantean los motivos de haber realizado estos cambios.

Proceso de Desarrollo.

- Trabajo de codificación acorde con la propuesta de herramientas, forma de organización, esquemas de monitoreo y bitácoras.



El modelo incremental es el indicado para nuestro proyecto ya que combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software que en nuestro proyecto sería (Servidor, cliente y rendering).

Bitácora

[illegible]

- Evaluación basada en objetivos (medibles) de la participación individual y de grupo.

NOMBRE	CONTRIBUCIÓN	ASISTENCIA	PORCENTAJE
ITZEL MOO	Idea principal, modularidad, proceso de desarrollo.	5/5	35%
ALBERTO MARÍN	Requerimientos, proceso de desarrollo.	5/5	35%
ERICK CÁRDENAS	Casos de uso, proceso de desarrollo.	5/5	30%

