|  |
| --- |
| **Anexo A. Ejemplo de Análisis de requisitos - Yakito** |

En este anexo se muestra un ejemplo de una lista de requisitos elaborada siguiendo las recomendaciones del estándar IEEE 830-1998.

La lista forma parte de la documentación del proyecto YakiTo, Conversor Multimedia, el cual, como su nombre lo indica, es un conversor de archivos multimedia libre y gratuito, que utiliza las bondades del proyecto FFMPEG para conseguir la mayor compatibilidad posible con la mayoría de formatos y códecs multimedia.

YakiTo es un proyecto de software libre, iniciado por Sergio García Mondaray, para el V Concurso Universitario de Software Libre.

**1. Restricciones de diseñ̃o**

RD.1 Las posibilidades de decodificación y codificación de YakiTo están limitadas por las posibilidades de FFMPEG. Por tanto, YakiTo sólo soportará los códecs compatibles con la biblioteca libavcodec, con la que trabaja FFMPEG.  (Lista de códecs compatibles en http://ffmpeg.org/general.html).

RD.2 Por limitaciones de FFMPEG, YakiTo no podrá soportar archivos de video con más de cinco canales de audio.

**2. Requisitos funcionales**

En esta sección se presentan los requisitos funcionales del sistema, los cuales están divididos en dos categorías: requisitos nominales y requisitos no nominales.

**2.1. Requisitos nominales**

FN.1 Descripción: La integración de FFMPEG debe ser completamente transparente al usuario, que no tiene por qué conocer qué es, para qué sirve, o cómo funciona dicho motor de recodificación.

Importancia: Esencial.

Validez:

* Medible: Si un usuario que desconozca FFMPEG es capaz de manipular la aplicación, convirtiendo vídeos satisfactoriamente, el requisito habrá sido cumplido.
* Alcanzable: Mediante la biblioteca JAVE (Java Audio Video Encoder), YakiTo podrá integrar FFMPEG en su propio ejecutable, evitando dependencias externas.
* Relevante: Se trata de un requisito fundamental en la identidad de YakiTo, así como en su sencillez para el usuario sin conocimientos. YakiTo debe mostrarse como una unidad sólida, sin dependencias de terceras aplicaciones.

FN.2 Descripción: Al iniciar la aplicación, YakiTo deberá revisar los directorios de idiomas y perfiles de conversión, para generar los elementos del menú en tiempo de ejecución (con el contenido de dichos directorios). Así pues, para que YakiTo ofrezca algún idioma adicional, bastaría con incluir el archivo adecuado en el directorio de idiomas y reiniciar la aplicación. Lo mismo ocurre con los perfiles de conversión. Este dinamismo convertirá a YakiTo en una aplicación muy fácilmente ampliable.

Importancia: Esencial

Validez:

* Medible: La carga de los archivos deberá realizarse correctamente, al iniciar la aplicación. Ademas sería conveniente añadir un ítem a algún menú para lanzar manualmente ese refresco, y poder así depurar la aplicación más fácilmente.
* Alcanzable: Se empleará archivos XML para almacenar los perfiles de conversión e idiomas de la interfaz.
* Relevante: Cumplir este requisito es fundamental para posibilitar las actualizaciones a través de internet, así como la ampliación de las posibilidades de la aplicación por parte del propio usuario.

**2.2. Requisitos no nominales**

FF.1 Descripción: En caso de que una conversión falle, YakiTo deberá detectar la situación y notificar al usuario, para preguntarle si desea continuar recodificando el resto de archivos o no.

Importancia: Condicional

Validez:

* Medible: Se tratará de hacer que una de las conversiones falle (intencionadamente, paralizando el hilo correspondiente), para estudiar si el comportamiento es el adecuado. YakiTo deberá percatarse del error y preguntar al usuario qué hacer.
* Alcanzable: Cada archivo será convertido por un hilo independiente. Además, YakiTo deberá tener el control absoluto de los hilos, para poder detener, pausar y continuar cualquiera de los hilos en cualquier momento.
* Relevante: El cumplimiento de este requisito mejoraría considerablemente la calidad del producto final. El poder detectar errores y saber controlar la situación haría de YakiTo una aplicación más robusta.

**3. Requisitos de interfaz**

IN.1 Descripción: La interfaz gráfica de YakiTo debe tener una lista con los archivos que se van a recodificar una vez que el usuario inicie la conversión. Además, la forma de añadir archivos a esa lista debe poder hacerse tanto mediante un botón destinado a tal efecto, como arrastrando archivos directamente desde el sistema operativo a la lista.

Importancia: Esencial.

Validez:

* Medible: Se añadirán archivos de las dos maneras, para asegurarse de que todo funciona correctamente.
* Alcanzable: Es sencilla la implementación de este requisito. Para ello se utilizará la clase FileDrop, creada por Robert Harder para dominio público.
* Relevante: Es un requisito importante para mejorar la experiencia de usuario, debido a que en la mayoría de aplicaciones comerciales que trabajan con varios ficheros, la técnica del *Drag & Drop* se ha vuelto tan común que los usuarios tienden a arrastrar ficheros de manera intuitiva.

IN.2   Descripción: La GUI debe ser lo más sencilla posible en primera instancia, para facilitar la tarea de conversión al usuario menos experimentado. Por otro lado, también debe ofrecerse la posibilidad de manipular muchos más parámetros de la recodificación, con el fin de satisfacer a los usuarios más exigentes y con experiencia.

 Importancia: Esencial.

Validez:

* Medible: Al ejecutar YakiTo se deberá apreciar el mínimo número de opciones necesarias para poder realizar una conversión de archivos multimedia exitosa. Pero se deberá tener también accesible una configuración avanzada.
* Alcanzable: En primer lugar la mejor manera de implementar este requisito sería mediante dos pestañas. La primera sería la mostrada por defecto, y con únicamente las opciones básicas; la otra pestaña sólo sería utilizada por usuarios experimentados. Así los controles complejos no son mostrados a no ser que el usuario quiera.
* Relevante: Este requisito es muy importante para facilitar al usuario menos experimentado la posibilidad de convertir sus archivos multimedia, sin apenas conocimientos, y sin llegar a sentirse sobrecargado por una interfaz llena de opciones.

IN.3   Descripción: En la lista de archivos a convertir, cada entrada, correspondiente a un único archivo, debe mostrar una barra de progreso individual, que muestre el porcentaje de conversión completada, así como el tiempo restante para terminar la conversión de ese archivo. Además, la GUI deberá mostrar en un lugar bien visible (por ejemplo la barra de estado) el tiempo restante total para finalizar la conversión de todos los archivos de la lista.

Importancia: Condicional

Validez:

* Medible: Se comprobará cómo aparece una barra de progreso por cada archivo añadido a la lista, así como el avance del progreso individual, y si corresponde el tiempo restante total con la suma de los tiempos individuales.
* Alcanzable: Se deberá manejar el paso del tiempo, para así poder estudiar el avance del progreso de conversión en un determinado periodo y poder estudiar el tiempo restante.
* Relevante: Este requisito, una vez cumplido, ofrecerá una mejora frente a otras aplicaciones parecidas, puesto que ninguna muestra una barra de progreso individual, ni tampoco ofrecen el tiempo restante, aunque resulta de gran utilidad.

IN.4   Descripción: La interfaz deberá dar la posibilidad al usuario de que seleccione uno de entre los dos modos de conversión posibles: secuencial y paralelo. La conversión secuencial convertirá uno a uno los archivos de la lista, por lo que el usuario deberá tener a su alcance la posibilidad de alterar el orden de los archivos de dicha lista. La conversión concurrente no requerirá de opciones adicionales, puesto que se limitará a convertir todos los archivos a la vez.

Importancia: Esencial

Validez:

* Medible: Se comprobará que ambos modos de conversión funcionan correctamente, así como el cálculo del tiempo restante (mencionado en el requisito IN.3) en ambos casos. El orden de los archivos para la conversión secuencial deberá ser fácilmente alterable por parte del usuario.
* Alcanzable: Para alterar el orden de conversión en el modo secuencial bastará con habilitar la posibilidad de arrastrar los archivos de dentro de la lista o, alternativamente, habilitar un par de botones que asciendan (den prioridad) o desciendan los archivos seleccionados en la lista.
* Relevante: El cumplimiento de este requisito supondrá una gran mejora de YakiTo respecto al resto de conversores de archivos multimedia.

IN.5   Descripción: La GUI deberá ofrecer la posibilidad de elegir de entre varios idiomas disponibles el que el usuario desee para la interfaz. Como introducir una opción dentro de un menú ofrecería una dificultad grande —si no entiendes el idioma actual, encontrar la opción del menú donde cambiarlo es muy difícil— se utilizará el extremo derecho de la barra de menú (rara vez utilizado) para mostrar el idioma actual de la interfaz con una bandera representativa, de tal modo que un clic sobre él hará que se despliegue una lista con el resto de idiomas disponibles (cada uno identificado por su bandera correspondiente). Hacer clic sobre uno de los idiomas de la lista lo activará directamente.

Importancia: Esencial.

Validez:

* Medible: Este requisito se verá satisfecho si el componente de idioma mencionado funciona correctamente y actualiza la interfaz con el idioma deseado (en un periodo de tiempo pequeño).
* Alcanzable: Se creará un menú en la barra de menús, destinado a la selección de idioma exclusivamente. Este menú se situará en el extremo derecho, alejándolo del resto de menús.
* Relevante: Este requisito constituye una diferencia abismal con el resto de alternativas, puesto que prácticamente ninguna ofrece la posibilidad de elegir de entre varios idiomas para la interfaz.

IN.6 Descripción: Todas las acciones ejecutables sobre la conversión deberán estar accesibles al usuario a través del menú en la aplicación. Pero, además, los controles más importantes: iniciar y abortar la conversión, deberán estar accesibles mediante botones en la parte inferior de la GUI; así se facilitará la conversión sin necesidad de usar el menú.

Importancia: Condicional.

Validez:

* Medible: El requisito se verá satisfecho si en la parte inferior de la GUI se muestran, con plena funcionalidad, los botones de iniciar y detener la conversión. Además sería conveniente deshabilitar cada uno cuando fuera conveniente —por ejemplo, no se puede abortar una conversión si no se ha iniciado todavía—.
* Alcanzable: Se crearán botones que ejecuten la misma acción que sus correspondientes ítems del menú.
* Relevante: Este requisito únicamente mejorará la experiencia de usuario, puesto que relegaría el menú a un segundo plano. Un usuario podría prescindir del menú si no necesita opciones avanzadas o la configuración de algún parámetro adicional.

**4. Requisitos de calidad**

CA.1 Descripción: YakiTo debe ser lo más sencillo posible, como ya se ha dicho, para posibilitar su utilización por parte de usuarios poco o nada experimentados con los formatos y códecs multimedia.

Importancia: Esencial.

Validez:

* Medible: Se estudiará la experiencia de diversos usuarios con pocos conocimientos de informática y formatos multimedia. Si no tienen dificultades para realizar varias conversiones, el requisito se verá satisfecho.
* Alcanzable: Se mantendrán dos pestañas en la interfaz, la de opciones básicas y la de opciones avanzadas. De esta manera, un usuario poco experimentado podrá emplear la primera pestaña, y realizar las conversiones de forma fácil (mediante el uso de perfiles de conversión); por otro lado, un usuario avanzado podrá, además, personalizar todos los parámetros de la recodificación en la segunda pestaña.
* Relevante: No superar este requisito haría caer la calidad de YakiTo, ya que una de sus principales características debe ser la facilidad de uso. Por tanto, este es uno de los requisitos más relevantes.

**5. Requisitos de evolución**

EV.1 Descripción: La aplicación debe estar disponible siempre en la última versión existente, así como ser debidamente actualizada en función de nuevas características que puedan ir surgiendo.

Importancia: Condicional.

Validez:

* Alcanzable: Para cumplir este requisito, la forja y el blog del proyecto deberán mantenerse vivos, informando de actualizaciones y novedades del proyecto.
* Relevante: Siempre resulta conveniente mantener un proyecto de software en desarrollo constante, sobre todo un proyecto de software libre. Aunque funcione bien la aplicación, deberá recibir mejoras periódicamente.

**6. Requisitos de proyecto**

PR.1 Descripción: Puesto que este proyecto se inicia para el Concurso Universitario de Software Libre, el proceso de desarrollo está acotado a varios meses. Aproximadamente se contará con siete meses para llevar a cabo el desarrollo inicial; tras ese periodo, YakiTo entrará en un proceso de mantenimiento constante (una vez haya terminado el concurso).

Importancia: Opcional.

Validez:

* Alcanzable: En el periodo especificado, el desarrollo de YakiTo deberá ser lo más completo posible (aunque el concurso no obliga a que sea completo). Se tratará de conseguir una versión usable, y en la que puedan cumplirse todos los requisitos esenciales.
* Relevante: Como se ha mencionado, no es necesario terminar el desarrollo de YakiTo en el periodo que indica la duración del concurso. Por tanto, este requisito pasa a un segundo plano.

**7. Requisitos de soporte**

SO.1 Descripción: El usuario final deberá contar con la máquina virtual de java instalada en su computador, con el fin de poder ejecutar YakiTo. YakiTo puede ser ejecutado en sistemas Windows, GNU/Linux y Mac OS.

Importancia: Esencial,

SO.2 Descripción: La computadora del usuario deberá contar con conexión a internet, con el fin de posibilitar la actualización de YakiTo con el servidor oficial. Sin esta característica, YakiTo no será actualizado, pero deberá ser plenamente funcional de todos modos.

Importancia: Opcional.

|  |
| --- |
| **Anexo B. Ejemplo de Análisis de requisitos – Centro Maestro Bienestar** |

## Restricciones de diseño

RD.1. El sistema será de tipo web.

RD.2. Se usará el servidor y el dominio que actualmente tiene la empresa.

RD.3. El sistema contará con dos tipos de usuarios, administrador tipo A y administrador tipo B.

RD.4. Se utilizará el *framework* Laravel para el desarrollo del sistema.

## Requisitos funcionales

Conjunto de requisitos que reflejan las funciones que debe prestar el sistema. Se clasifican en las siguientes subsecciones:

### Requisitos funcionales nominales

|  |  |  |
| --- | --- | --- |
| Título: | | FN 1- Módulo de Reservaciones |
| Descripción: | | Módulo en el cual el administrador podrá ver la información de las reservaciones, el administrador podrá dar de alta, ver información, aprobar y cancelar reservaciones también podrá realizar la factura de la reservación. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El módulo permitirá al administrador ingresar los datos de la reservación, como son el nombre de la persona, la sección del SNTE a la que pertenece, la fecha de la reservación, el método de pago el cual mostrará varias opciones como pago con tarjeta, transferencia, efectivo etc., y los datos de hasta 10 acompañantes y registrarlos en la base de datos, también podrá revisar el estado del pago que puede ser pendiente, pagado, etc., aprobar, cancelar o modificar reservaciones y cambiar el status del pago. Las reservaciones canceladas se eliminan de forma lógica en la base de datos y se podrán consultar en el apartado de reservaciones canceladas, podrá realizar modificaciones a la reservación, como la fecha de la misma o agregar más acompañantes y generar una factura en PDF.  Todos los movimientos realizados en el módulo de reservaciones se agregarán a un log de actividades junto con el usuario que realizo el movimiento. |
| **Alcanzable:** | A través de un formulario dividido en secciones el administrador podrá ingresar la información de la reservación y los acompañantes, el administrador también podrá agregar la información de facturación a través de otro formulario, la información de la reservación, el usuario, facturación y los acompañantes se agregará a la base de datos ocupando las entidades y el controlador de la base de datos de Laravel.  Una vez agregada la reservación el administrador tendrá que aprobar la reservación a través de un botón y un *modal* de confirmación, utilizando la entidad “reservación” se actualizará el estado de la reservación en la base de datos así mismo para actualizar el estatus de pago el administrador utilizara un *dropdown* para seleccionar el estatus y un *modal* de confirmación, el sistema realizara la actualización a través de la entidad “pago”.  Si se requiere cancelar una reservación, el administrador podrá realizar este movimiento a través de un botón y un *modal* que mostrará un *dropdown* con las causas de cancelación.  Cuando la reservación esté completa a través de la entidad internos el sistema creará los nuevos internos.  Utilizando la librería Dompdf se generará un *PDF* con la factura de la reservación. |
| **Relevante:** | Al cumplir con este requisito el administrador podrá realizar la gestión de las reservaciones del recinto, las cuales son esenciales para el sistema. |

|  |  |  |
| --- | --- | --- |
| Título: | | FN 2- Módulo de Internos |
| Descripción: | | Una vez que se aprueba la reservación y el pago, el usuario pasa a ser un interno, el administrador será capaz de dar de alta, eliminar y modificar la información de los internos, y asignarles una cama, también se mostrará la información más relevante del interno con opción de ver su información completa, las compras que ha hecho y los servicios que ha consumido. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El sistema moverá automáticamente a los usuarios de la reservación a internos, el módulo permitirá al administrador agregar y ver la información relevante de los internos y a través de una lista desplegable asignarles una cama que esté disponible también contará con las opciones de ver información completa, ver compras y ver servicios consumidos. |
| **Alcanzable:** | El administrador podrá ver la información de los internos a través de una tabla, en la tabla aparecerán con botones las opciones que el administrador podrá realizar con los internos.  A través de un *dropdown* el administrador podrá asignarles una cama a los internos, el administrador podrá quitar la cama a un interno a través de un botón, el sistema a través de la entidad “Interno” y “Cama” agregará o modificará la información de la base de datos,  A través de un formulario el administrador podrá agregar o modificar la información de los internos, el sistema ocupará las entidades o el controlador de base de datos para realizar los cambios correspondientes. |
| **Relevante:** | Al cumplir con este requisito el administrador podrá realizar la gestión de los internos del recinto que es esencial para el sistema. |

|  |  |  |
| --- | --- | --- |
| Título: | | FN 3- Módulo de inventario |
| Descripción: | | Módulo en el cual el administrador podrá dar de alta, dar de baja, realizar búsquedas y modificar un producto. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El módulo permitirá ingresar los datos de un producto, el cual se registrará en la base de datos, así mismo podrá seleccionarlo y eliminarlo, o bien podrá realizar una búsqueda del producto. También podrá seleccionar un producto y modificar sus datos o el stock disponible del producto. |
| **Alcanzable:** | A través de un formulario el administrador podrá agregar o modificar un producto del inventario, el sistema realizará la acción correspondiente utilizando la entidad “Producto”.  El administrador podrá ver la información de los productos a través de una tabla con información de los mismos, cada producto tendrá un botón para modificar y eliminar, al dar clic en el botón de eliminar se desplegará un *modal* de confirmación, y a través de la entidad “Producto” se realizará la eliminación lógica del producto. |
| **Relevante:** | Al cumplir este requisito el administrador podrá realizar la gestión completa de los productos del recinto. |

|  |  |  |
| --- | --- | --- |
| Título: | | FN 4- Módulo de servicios |
| Descripción: | | Módulo en el cual el administrador podrá dar de alta, dar de baja, realizar búsquedas y modificar un servicio. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El módulo permitirá ingresar los datos de un servicio, el cual se registrará en la base de datos, así mismo podrá seleccionarlo y eliminarlo, o bien podrá realizar una búsqueda. También podrá seleccionar un servicio y modificar sus datos. |
| **Alcanzable:** | A través de un formulario el administrador podrá agregar o modificar un servicio, el sistema realizará la acción correspondiente utilizando la entidad “Servicio”.  El administrador podrá ver la información de los servicios a través de una tabla, cada servicio tendrá un botón para modificar y eliminar, al dar clic en el botón de eliminar se desplegará un *modal* de confirmación, y a través de la entidad “Servicio” se realizará la eliminación lógica del producto. |
| **Relevante:** | Al cumplir este requisito el administrador podrá realizar la gestión completa de los servicios del recinto. |

|  |  |  |
| --- | --- | --- |
| Título: | | FN 5- Módulo de Usuarios |
| Descripción: | | Módulo en el cual el administrador podrá dar de alta, dar de baja, ver y modificar un usuario. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El módulo permitirá ingresar los datos de un usuario y el tipo de usuario, el cual se guardará en la base de datos, así mismo podrá seleccionarlo para eliminarlo o modificar sus datos. Al momento de eliminar y modificar se desplegará un mensaje de confirmación. |
| **Alcanzable:** | A través de un formulario el administrador podrá dar de alta o modificar un administrador, utilizando una tabla se mostrarán los administradores actuales con un botón para modificar y otro para eliminar, al eliminar un administrador se mostrará un *modal* de confirmación, el sistema utilizando la entidad “*User*” realizará las acciones correspondientes. |
| **Relevante:** | Al cumplir este requisito se otorgará la gestión de los usuarios que pueden operar el sistema, la cual es necesaria para poder desempeñar las demás funciones del sistema. |

|  |  |  |
| --- | --- | --- |
| Título: | | FN 6- Módulo de Noticias |
| Descripción: | | Módulo en el cual el administrador podrá dar de alta, dar de baja, y modificar las noticias que se presentan en la página principal. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El módulo permitirá ingresar el título y cuerpo de la noticia, el cual se registrará en la base de datos, así mismo podrá eliminarla o modificar sus datos. |
| **Alcanzable:** | Utilizando un formulario el administrador podrá crear o modificar una noticia, las noticias se mostrarán en una tabla donde se podrá ver la imagen, el título y el *abstract*, al eliminar la noticia el sistema mostrará un *modal* de confirmación, utilizando la entidad “Noticia” y el sistema de archivos de Laravel se realizarán las acciones correspondientes. |
| **Relevante:** | Al cumplir este requisito el administrador podrá realizar la gestión completa de las noticias del recinto. |

|  |  |  |
| --- | --- | --- |
| Título: | | FN 7- Módulo de Ventas |
| Descripción: | | Módulo en el cual el administrador podrá realizar una venta de un producto o servicio a un interno y generar un comprobante personalizado de la compra. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El módulo permitirá seleccionar un interno para realizar la venta de un producto o servicio, a través de una lista se mostrarán los productos y los servicios disponibles, el administrador podrá seleccionar varios productos o servicios y la forma de pago, en el caso de los productos la venta del producto se verá reflejada en el stock actual del mismo, se creará un comprobante de compra en PDF personalizado con la información del interno. |
| **Alcanzable:** | Los internos tendrán un botón para realizar una venta de productos o servicios, a través de dos tablas en la primer tabla se mostrarán todos los productos o servicios actuales, en esta tabla el administrador utilizando un *spinner* podrá seleccionar la cantidad de productos y agregarlos a través de un botón a la segunda tabla donde se muestran todos los productos o servicios que el interno adquirirá, una vez realizado esto el administrador podrá seleccionar el tipo de pago que se mostraran en un *dropdown* y realizar la venta con un botón,el sistema utilizara la entidad “Venta”, “Producto” o “Servicio” para realizar las acciones correspondientes y utilizando la librería Dompdf mostrar el comprobante de compra en PDF. |
| **Relevante:** | Al cumplir este requisito el administrador podrá realizar la gestión completa de las ventas del recinto. |

|  |  |  |
| --- | --- | --- |
| Título: | | FN 8- Módulo de Estadísticas |
| Descripción: | | Módulo en el cual el administrador podrá ver las estadísticas del recinto. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El módulo permitirá ingresar al administrador una fecha o un rango de fechas y a partir de consultas a la base de datos, le mostrará gráficas con las estadísticas de las ventas y reservaciones del recinto, como son el producto más vendido, el número de reservaciones, la sección SNTE que más se hospedó etc. |
| **Alcanzable:** | Utilizando la librería *JQuery.Charts* y consultas a la base de datos utilizando el controlador de base de datos de Laravel se mostrarán las gráficas correspondientes. |
| **Relevante:** | Al cumplir este requisito se otorgará al administrador una herramienta que ayude a la toma de decisiones de la empresa. |

### Requisitos funcionales no nominales

|  |  |  |
| --- | --- | --- |
| Título: | | FF 1- Error de inicio de sesión |
| Descripción: | | El sistema no permitirá acceder a usuarios no registrados, en dado caso que se ingrese un usuario o contraseña incorrecta se mostrará un mensaje de error. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | Al ingresar un usuario erróneo el sistema mostrará un mensaje de error y no permitirá acceder al sistema. |
| **Alcanzable:** | Al ingresar un usuario y contraseña el sistema a través del controlador de autenticación realizará la consulta a la base de datos, en caso de que el usuario y/o contraseña no existan o no correspondan, el sistema mostrará un mensaje de error. |
| **Relevante:** | El sistema será capaz de protegerse contra usuarios no registrados. |

|  |  |  |
| --- | --- | --- |
| Título: | | FF 2- Error de conexión a la base de datos |
| Descripción: | | El sistema mostrará un error de conexión. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | Al tratar de realizar una consulta a la base de datos el sistema mostrará un error. |
| **Alcanzable:** | El *framework* Laravel cuenta con sistema errores donde ya se controla este error, al fallar la conexión a la base de datos Laravel muestra una pantalla de error. |
| **Relevante:** | El sistema evitará pérdidas de datos. |

|  |  |  |
| --- | --- | --- |
| Título: | | FF 3- Error de conexión al servidor |
| Descripción: | | Si la conexión al servidor no está disponible se mostrará un error. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | Al tratar de acceder al sistema el explorador mostrará un mensaje de error HTTP. |
| **Alcanzable:** | Los Navegadores web muestran un error al no poder acceder al sitio web, donde muestran el código de error HTTP o si no se logró establecer conexión. |
| **Relevante:** | El recinto se podrá dar cuenta que algo anda mal con el sistema. |

|  |  |  |
| --- | --- | --- |
| Título: | | FF 4 - Falta de archivos |
| Descripción: | | Si el sistema no logra encontrar un archivo mostrara un error. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | Al tratar de acceder a un archivo, como el de bienvenida al interno, el sistema mostrará un error. |
| **Alcanzable:** | Al verificar si se logró obtener el archivo se podrá mostrar una pantalla de error en caso de fallo. |
| **Relevante:** | El administrador podrá seguir utilizando el sistema y realizar alguna acción para corregir el error. |

|  |  |  |
| --- | --- | --- |
| Título: | | FF 5 – Error al generar archivo |
| Descripción: | | Si el sistema no logra generar un archivo mostrara un error. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | Al tratar de generar a un archivo, como el comprobante de venta, el sistema mostrará un error. |
| **Alcanzable:** | Ya que los archivos se crean a través de una plantilla HTML, Laravel podrá controlar este error y mostrar una pantalla con el código de error. |
| **Relevante:** | El administrador podrá seguir utilizando el sistema y realizar alguna acción para corregir el error. |

## Requisitos de interfaz

|  |  |  |
| --- | --- | --- |
| Título: | | IN 1 – Colores de la empresa |
| Descripción: | | La paleta de colores del sistema tendrá relación con los colores de la empresa. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | Los colores del sistema tendrán relación con los colores del logo y de la empresa. |
| **Alcanzable:** | Utilizando el programa Photoshop se podrá obtener el código de los colores del logo de la empresa y utilizarlos en la interfaz del sistema. |
| **Relevante:** | Al utilizarse los colores de la empresa para el sistema, el sistema seguirá la imagen de la empresa. |

|  |  |  |
| --- | --- | --- |
| Título: | | IN 1 – Estándares de diseño de interfaz de usuario. |
| Descripción: | | El sistema será lo más intuitivo posible para el usuario. |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El sistema seguirá los estándares marcados por la W3C para el desarrollo de interfaces gráficas |
| **Alcanzable:** | Al pasar la prueba de interfaz de la página oficial de W3C se podrá verificar la implementación de dichos estándares. |
| **Relevante:** | La interfaz de usuario del sistema cumplirá con los estándares de la W3C y será más sencillo para los usuarios el uso del sistema. |

## Requisitos de calidad

|  |  |  |
| --- | --- | --- |
| Título: | | CA.1 estándares de programación en PHP PSR-2 y PSR-4. |
| Descripción: | | El sistema seguirá los estándares de programación PSR-2 y PSR-4 |
| Importancia: | | Esencial. |
| Validez: | **Medible:** | El código del sistema seguirá los estándares de programación en PHP PSR-2 y PSR-4, ya que el uso del *framework* Laravel promueve el uso de estos. |
| **Alcanzable:** | Al ver el código del sistema se podrán observar las buenas prácticas de programación relacionadas con el estándar PSR-2 y PSR-4. |
| **Relevante:** | Será más sencillo de entender el código para el mantenimiento o actualización del sistema. |