**Contenidos**

[1.Introducción 2](#_gjdgxs)

[2.Requisitos funcionales del sistema 2](#_30j0zll)

[3.Cualidades del Sistema 2](#_1fob9te)

[3.1Usabilidad 2](#_3znysh7)

[3.2Fiabilidad 2](#_2et92p0)

3.3Rendimiento 2

[3.4Mantenibilidad 3](#_tyjcwt)

3.5Seguridad 3

[4.Interfaces del Sistema 3](#_3dy6vkm)

[4.1Interfaces de Usuario 3](#_1t3h5sf)

[4.1.1Aspecto 3](#_4d34og8)

[4.1.2Requisitos de diseño y navegación 3](#_2s8eyo1)

[4.1.3Consistencia 4](#_17dp8vu)

[4.1.4Requisitos de Personalización de Usuario y Customizacion 4](#_3rdcrjn)

[4.2Interfaces con sistemas o dispositivos externos 4](#_26in1rg)

[4.2.1Interfaces de Software 4](#_lnxbz9)

4.2.2Interfaces de Hardware 4

[4.2.3Interfaces de Comunicaciones 4](#_35nkun2)

[5.Reglas del Negocio 5](#_1ksv4uv)

[5.1<Nombre de Clases de Regla> 5](#_44sinio)

[5.1.1<Nombre de Regla e ID> 5](#_2jxsxqh)

[6.Restricciones del Sistema 5](#_z337ya)

[7.Conformidad del Sistema 5](#_3j2qqm3)

[7.1Requisitos de Licencia 5](#_1y810tw)

[7.2Aviso Legal, Derechos de Autor y Otros Avisos 5](#_4i7ojhp)

[8.Documentación del Sistema 5](#_2xcytpi)

Sistema Web para Gestión de Fotografías

**Requisitos a nivel de sistema**

# Introducción

Este documento se encarga de describir los requisitos funcionales no contemplados en los casos de uso, cualidades, interfaces y dependencias de hardware, software y comunicaciones del sistema, además de las reglas del negocio.

# Requisitos funcionales transversales

El sistema contará con funcionalidades tales como:

* El sistema registrará las solicitudes hechas por los Clientes para servicios de fotografía, permitiendo llevar un historial de las solicitudes hechas por estos.
* El sistema gestionará a todos los usuarios tanto operadores del sistema como los beneficiarios del servicio de prestamos.
* El sistema gestionará el ingreso de libros nuevos, reposiciones y descarte de estos.

# Cualidades del Sistema

## Usabilidad

* El sistema deberá ser fácil de aprender, para ello un usuario experimentado debe ser capaz de utilizar todas las funciones del sistema tras un entrenamiento de 2 horas, tras el cual no cometerá más de 3 errores diarios en media.
* El sistema tendrá un diseño intuitivo y simple brindado por el framework angular material basado en componentes.

## Fiabilidad

* El sistema estará disponible y accesible un 90% del tiempo semanal.
* El sistema será capaz de restablecer el estado deseado frente a algún fallo apoyándose en el sistema controlador de versiones “Git”.

## Rendimiento.

* El sistema responderá con un tiempo de respuesta no mayor a 2 segundos por operación en el entorno local.

## Soportabilidad

* El sistema estará desarrollado en clases que a su vez serán subdividos, para que algún cambio en alguna funcionalidad tenga un impacto mínimo al sistema.
* El sistema tendrá clases/métodos que podrán ser reusados.
* El sistema manejará los estándares psr2 (http://www.php-fig.org/psr/psr-2/) y psr4 (http://www.php-fig.org/psr/psr-4/) para el código de programación que permitirá su futuro mantenimiento o análisis de errores.
* El sistema será fácil de instalar, para ello tendrá una guía de configuración e instalación en su repositorio alojado en gitlab.

# Interfaces del Sistema

## Interfaz de Usuario

### Aspecto

* El sistema presentará el look and feel en ventanas, colores e íconos basandose en la plataforma web institucional de la universidad.
* El sistema mostrará formatos y tamaños: Títulos tamaño 36 tipo Arial, textos tamaño10 Calibrí y botones tamaño 10 tipo Arial.
* EL sistema deberá poseer mensajes de error orientados al usuario final.
* El sistema seguirá un aspecto homogéneo de forma que se mantenga un conjunto coherente de colores, fuentes y estilos para la aplicación.

### Requisitos de distribución y navegación

* El sistema presentará las áreas principales de la pantalla conforme a los estándares web de la biblioteca central de la universidad.

### Consistencia

* Los requisitos para los controles de navegación, tamaños y formas de áreas de pantallas se mantendrán conforme a los estándares web de la universidad.

### Requisitos de Personalización de Usuario y Customización

* El sistema no permitirá personalizar el contenido mostrado a los usuarios.

## Interfaces a sistemas externos o dispositivos

* El sistema no permitirá interactuar con otros sistemas externos, ya que no se cuenta con el acceso actualmente a estos.

### Interfaces de Software

* El sistema funcionará sobre una base de datos MySql.

### Interfaces de Hardware.

* El sistema estará implementado inicialmente en el localhost del equipo, 127.0.0.1, puerto 3000

### Interfaces de Comunicaciones

* El sistema contendrá interfaces de comunicación con los estándares Web y fundamentalmente se deben basar en protocolos HTTP, HTTPS para la comunicación con usuarios finales y para desarrollo de Web Services REST.
* El sistema necesitará a futuro la utilización de otros protocolos que complementan las diferentes interfaces de comunicación entre cada uno de los componentes que deberán ser definidos en un nivel mayor de diseño arquitectónico.

# Reglas del Negocio

# Restricciones del Sistema

* El sistema estará limitado a guardar una base de datos, debido a los pocos recursos con los que se cuenta se optó por usar MySql como gestor de la base de datos.
* El sistema contará con un servidor cuyo disco duro de 1 TB, memoría RAM de 4GB.
* La API será desarrollada en el framework Laravel de PHP al ser de fácil aprendizaje y contar con componentes prefabricados como el de autenticación por OAuth2 haciendo uso de su paquete Passport de Laravel. Además el framework provee una estructura conveniente para la aplicación:
  + Uso de componentes Request que ayudan a separar la lógica de validación de los controladores.
  + Separación de las diferentes rutas de la aplicación según convenga.
  + Uso de su ORM llamado Eloquent para realizar el mapeo con la base de datos.
  + Manejo de migraciones mediante componentes Migration para llevar un registro histórico de los cambios que se realice a la base de datos estructuralmente.
  + Uso de componentes Seeder que puede servir para dar una data inicial al sistema o para pruebas.
  + Herramientas de testing sobre API REST en el formato JSON. Permite realizar pruebas unitarias así como pruebas de aceptación.
* La Aplicación Web sera desarrollada en AngularJS al contar con una estructura de bajo acoplamiento.

# Conformidad del Sistema

## Requisitos de Licencia

* El sistema usará componentes libres de licencia para su desarrollo e implementación.
* El sistema tendrá aplicará las restricciones de terceros que usen el software.

## Aviso Legal, Derechos de Autor y Otros Avisos

* El sistema no estará sujeto a derechos de autor, siendo este de caracter libre adaptándose a la licencia pública general de GNU o LGPL garantizando la libertad de compartir y modificar el sistema.

# Documentación del Sistema

* La documentación para usuarios en línea estará dada por el grupo del proyecto, siendo el lider del proyecto el responsable de esta, será de acceso libre.
* El sistema no tendrá inicialmente un soporte en línea, pudiendo esto ser modificable en el futuro dependiendo de la necesidad del usuario.