|  |
| --- |
| **Proyecto Red Social de Imágenes** |

|  |  |  |
| --- | --- | --- |
| Fecha de Rev.: | Versión 1.0 |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Plan de proyecto | | |
| Versión | **Fecha**  **Versión** | **Descripción del Cambio** | **Aprobado Por** |
| 01 | 25/11/2020 | Creación de documento | DB |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Contenido

[Contenido 2](#_heading=h.gjdgxs)

[1. Información del proyecto 3](#_heading=h.30j0zll)

[2. Descripción del proyecto 3](#_heading=h.3znysh7)

[3. Asunciones 3](#_heading=h.tyjcwt)

[4. Restricciones 3](#_heading=h.3dy6vkm)

[5. Requisitos de ley 4](#_heading=h.4d34og8)

[6. Organización del Proyecto 4](#_heading=h.17dp8vu)

[7. Conocimiento y Habilidades Requeridas 5](#_heading=h.3rdcrjn)

[8. Equipo de trabajo asignado al proyecto 6](#_heading=h.26in1rg)

[9. Ambientes definidos para el proyecto 6](#_heading=h.lnxbz9)

[10. Necesidades tecnológicas 7](#_heading=h.35nkun2)

[11. Seguimiento 7](#_heading=h.1ksv4uv)

[12. Fases del ciclo de vida del proyecto 8](#_heading=h.44sinio)

[13. Actividades de Cierre 9](#_heading=h.2jxsxqh)

[14. Backlog 9](#_heading=h.z337ya)

# Información del proyecto

|  |  |
| --- | --- |
| **Nombre del Proyecto** | **Red Social de Imágenes** |
| **Breve descripción del proyecto** | Red social desarrollada para profesionales y Amateur para postear imágenes. |
| **Cliente** | **Cliente Final:** Deivin Berrio |
| **Duración** | 30 días |
| **Responsable del proyecto (Dev.)** | Dairis Bello |
| **Responsable del proyecto (Cliente)** | Gabriela Franco |
|  |  |

# Descripción del proyecto

La solución integral ofrecida se denomina Red Social de Imágenes, la cual permite la creación, edición, visualización y búsqueda de imágenes.

Esta solución tiene la capacidad de responder a los niveles de eficiencia requeridos en el manejo de altos volúmenes de imágenes. La infraestructura que soporta la aplicación presenta alta disponibilidad en línea.

# Asunciones

* No hay.

# Restricciones

* Entregar una versión de la Red social en completo funcionamiento en Diciembre/28

# Requisitos de ley

* No hay

# Organización del Proyecto

El siguiente gráfico muestra la organización del proyecto y las líneas de autoridad definidas para el mismo.

|  |
| --- |
|  |
| Cliente y/ Product Owner  Scrum Master  Team Scrum  Desarrolladores  QA  Requerimiento |
| **Organización del proyecto** |

# Conocimiento y Habilidades Requeridas

A continuación, se lista los conocimientos y/o habilidades necesarias para el proyecto.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Conocimiento y/o Habilidad | Recurso (Numero de recursos que requieren el conocimiento) | Fecha planeada (Fecha en la que se requiere el conocimiento y/o habilidad) | Fecha Real (Fecha en la que se adquiere el conocimiento) | Forma de obtención (Capacitación, Auto-estudio, Contratación) | Observación |
| Scrum | 5 | 23/11/2020 | Noviembre | Capacitación |  |
| Java | 5 | 23/11/2020 | Noviembre | Capacitación |  |
| SQL | 5 | 23/11/2020 | Noviembre | Capacitación |  |
| Python | 5 | 23/11/2020 | Noviembre | Capacitación |  |

A continuación, se describen los conocimientos y habilidades requeridas para los diferentes roles de los integrantes del equipo del proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| Cargo | # personas | Habilidades y clasificación | Disponibilidad requerida |
| Product Owner | 1 | Product Owner | 100% |
| Scrum Master | 1 | Scrum Master | 100% |
| Desarrolladores Junior | 2 | SQL , JAVA, PYTHON | 100% |
| Documentador/Pruebas | 1 | SQL, JAVA, PYTHON | 100% |

# Equipo de trabajo asignado al proyecto

A continuación, se describen los conocimientos y habilidades requeridas para los diferentes roles de los integrantes del proyecto:

|  |  |  |  |
| --- | --- | --- | --- |
| Cargo | Nombres y Apellidos | Responsabilidades | % Asignado |
| Product Owner | Gabriela Franco | Identificar e indicar el “que” debe tener el producto. | 100% |
| Scrum Master | Dairis Bello | Remover impedimentos. | 100% |
| Desarrolladores | * Lismar Alvear * Juan Velez | Desarrollo de Software | 100% |
| QA | Doriana Rolong | Documentación y Pruebas | 100% |

# Ambientes definidos para el proyecto

A continuación, se describen los ambientes definidos para el proyecto, teniendo en cuanta que las necesidades tecnológicas para cada ambiente son las mismas descritas en el ítem “Necesidades tecnológicas”.

|  |  |  |  |
| --- | --- | --- | --- |
| Ambiente | Descripción | URL | Responsable |
| DEV | Ambiente utilizado para desarrollar el producto |  | Scrum Master |
| QA | Ambiente utilizado para la verificación del producto |  | Scrum Master |
| Producción | Ambiente productivo. |  | Scrum Master |

# Necesidades tecnológicas

Este ítem busca describir las necesidades tecnológicas en cuanto a hardware y software se refiere.

|  |  |  |
| --- | --- | --- |
| Herramientas/ Equipos/ Software/ Capacitaciones | Cantidad | Comentarios |
| Base de datos SQL | 2 | Almacena los documentos e información correspondientes a la Red Social de Imágenes |
| Website | 1 | Contiene los webs sites para:   * Red Social de Imágenes |
| Python | 1 | Desarrollar Software |
| Java | 1 | Desarrollar Software |

# Seguimiento

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Compromiso | Periodicidad | Objetivo | Responsables | Fecha Planeada | Fecha Real | Evidencia |
| Avance estado del proyecto | Diario | Informar sobre el cumplimiento de la programación diaria y los temas atendidos | Product Owner | 4:20pm | 4:20pm  . | Reporte de Planificación de Tareas para el sprint. |
| Requerimientos | Antes de planeación de nuevo sprints | Seleccionar los requerimientos que se trabajarán en el siguiente sprint y la definición detallada de los mismos. | Product Owner | Sujeto a fechas de inicio y fin de cada sprint | Sujeto a fechas de inicio y fin de cada sprint | Requerimientos actualizados |

# Fases del ciclo de vida del proyecto

En este ítem se detalla la metodología, las herramientas y los responsables asociados a cada una de las fases del ciclo de vida del proceso de diseño y desarrollo que se ejecutarán a lo largo del proyecto. Estas fases se desarrollarán usando el **modelo iterativo** y a su vez los ciclos de trabajo o sprint definidos en la metodología SCRUM.

Cada iteración y/o ciclo tienen una duración de 8 días, vale la pena aclarar que, debido a condiciones especiales, en cuanto a requerimientos, días festivos locales y/o nacionales, entre otros. La duración del Sprint puede variar.

|  |  |  |  |
| --- | --- | --- | --- |
| Fase del ciclo de vida | | Metodología | Responsable |
| Diseño y Desarrollo | **Planeación ciclo incremental** | Scrum | Product Owner |
| **Diseño** | Scrum | Product Owner y Scrum team |
| **Desarrollo** | Scrum | Equipo de Desarrollo |
| **Pruebas** | Scrum | Equipo de Pruebas y documentación |
| **Documentación** | Scrum | Equipo de Pruebas y documentación |
| **Seguimiento y control del ciclo incremental** |  | Product Owner, Equipo de Desarrollo, Equipo de Pruebas y documentación, interesados. |

# Actividades de Cierre

Las actividades de cierre que aplican al proyecto se describen a continuación:

|  |  |  |
| --- | --- | --- |
| Actividad de Cierre | Responsable | Evidencia |
| Cumplimiento de las expectativas del proyecto | Product Owner/cliente | Acta de reunión |
| Acta de Aceptación final | Product Owner/cliente | Acta de reunión |

# Backlog

A continuación el Backlog con la planeación de los 4 Sprint.

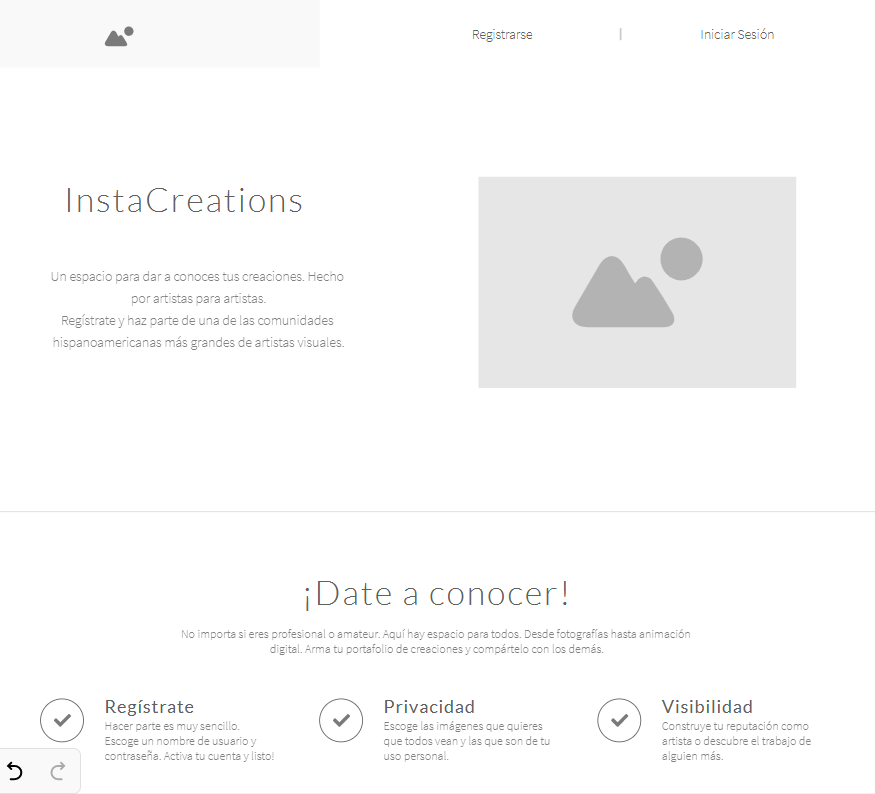
[PRODUCT BACKLOG](https://drive.google.com/file/d/13bYuqX6SS661tcFmxwZGZ0ujHcsNoFFX/view?usp=sharing)

# Requisitos Funcionales y wireframes

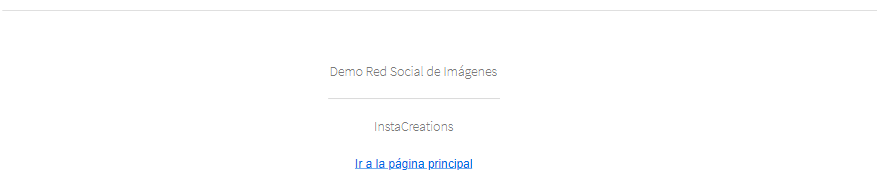
**Diagrama de Casos de Uso**

****

**VISTA INICIAL**

****

****

****

**REQUERIMIENTO 1**

Registrar usuarios. Un interesado se debe registrar ingresando un usuario, contraseña y correo electrónico. Debe haber un proceso de activación. Por ejemplo, una vez que el usuario se haya registrado, éste debe recibir un correo electrónico con un enlace para activar su cuenta.

**Especificación de requerimientos para registro de usuario**

|  |  |
| --- | --- |
| **Título** | **Registro de usuario** |
| **Prioridad** | **Alta** |
| **Diseño de la página** |  |
| **Especificación de requerimientos** | Para el ingreso a la red social de imágenes se requiere una página que permita registrar los usuarios que van a utilizar la red social. Para acceder al formulario de registro, en la página principal, ubique el botón *verde* que dice “Regístrese aquí”. Esta página debe contener:  Nombre de usuario – Se debe especificar el nombre con el cual se identificará al usuario en el sistema. Cuadro de texto que permite diligenciar hasta 50 caracteres. Campo requerido. Si existe un usuario registrado con el mismo nombre de usuario mostrar una advertencia de que el nombre de usuario ya existe y debe cambiarse.  Contraseña - El usuario debe especificar una contraseña que tenga como mínimo 8 caracteres, entre mayúsculas, minúsculas y un carácter específico, si la contraseña no cumple con los requisitos mínimos el sistema debe enviar una advertencia de que la contraseña no cumple con los requerimientos. Campo requerido.  Email – Campo de texto que permite registrar el correo electrónico de un usuario a donde se le enviarán notificaciones. Este campo debe cumplir con la siguiente estructura “XXX@XXX.XXX”, si el correo no cumple con la estructura, el sistema debe enviar una advertencia diciendo que el correo no cumple con los requisitos mínimos. Campo requerido.  Botón cancelar – Botón que permite cancelar el registro del usuario y redirige a la página de inicio.  Botón guardar – Botón que permite registrar al usuario en el sistema. Valida que los campos requeridos se encuentren diligenciados, cuenten con la estructura definida y que el correo no esté asociado a una cuenta existente. Si las validaciones son exitosas mostrar mensaje indicado que una notificación ha sido enviada a su correo electrónico con un link para la activación de su cuenta. Debe generar el envío de un correo electrónico con un enlace para la activación de la cuenta. |
| **Criterios de aceptación** | * El registro de usuario solicita nombre, contraseña y el correocomo requerido. * El campo “usuario” recibe hasta 50 caracteres. * El campo “usuario” no se encuentra repetido en el sistema. * El campo contraseña permite mínimo 8 caracteres alfanuméricos, incluidos una mayúscula, una minúscula y un carácter específico. * El campo Email tiene la siguiente estructura XXX@XXX.XXX. * Al guardar el sistema comprueba que el correo no esté asociado a una cuenta existente. * Al cancelar el sistema le muestra al usuario la página principal de la red social. * Al guardar el usuario satisfactoriamente se envía una notificación al correo electrónico registrado con un enlace para la activación de la cuenta. |

|  |  |
| --- | --- |
| **Título** | **Activación de cuenta** |
| **Prioridad** | **Media** |
| **Diseño de la página** |  |
| **Especificación de requerimientos** | El usuario recibirá una notificación a su correo electrónico que contendrá un enlace para activar su cuenta. Al hacer *click* en el enlace éste será redirigido a una página donde muestre un mensaje “Activación de cuenta exitosa”. |
| **Criterios de aceptación** | Al hacer click en el enlace éste redirige a una página con mensaje “Activación de cuenta exitosa”. |

**REQUERIMIENTO 2**

Proveer un portal de acceso, donde los usuarios puedan acceder a la plataforma, si se autentican, usando usuario y contraseña, exitosamente. Esto debe cumplir con los requerimientos mínimos de seguridad.

**Especificación de requerimientos para inicio de sesión**

|  |  |
| --- | --- |
| **Título** | **Iniciar sesión** |
| **Prioridad** | **Alta** |
| **Diseño de la página** | **Inicio de sesión**    **Sesión principal** |
| **Especificación de requerimientos** | Para el ingreso al sistema de chat se requiere una página que permita iniciar sesión al usuario, esta página tiene relevancia una vez el usuario haya realizado el registro, la página iniciar sesión contiene lo siguiente:  Nombre de usuario - Campo donde se debe especificar el nombre con el que se identifica el usuario, el cual fue asignado previamente en el registro.  Si el nombre de usuario no se encuentra registrado debe mostrar una advertencia y sugerirle al usuario que realice el registro, mostrándole la opción para redireccionarlo a la página correspondiente.  Contraseña - Campo donde se especifica la contraseña que el usuario previamente estableció en la página de registro.  Si la contraseña ingresada es incorrecta se debe mostrar un mensaje tipo pop-up, donde se le informe al usuario que la contraseña ingresada es incorrecta.  Botón Ingresar: Una vez los datos ingresados en los campos nombre de usuario y contraseña sean verificados con los registrados, se pasa a la página principal de la red social de imágenes.  Botón Restablecer Contraseña: En caso de que el usuario olvide su contraseña, presionará el botón restablecer que lo enviará a la página del restablecimiento de contraseña. |
| **Criterios de aceptación** | La página Inicio de sesión solicita el nombre de usuario y la contraseña.  Al presionar el botón ingresar, el sistema valida que tanto el nombre de usuario como la contraseña sean correctos.  En el caso que el nombre de usuario o la contraseña sean incorrecto se muestra un mensaje indicando el error al usuario. |

**REQUERIMIENTO 3**

Ofrecer la opción para recuperar la contraseña en caso de olvido para los usuarios.

**Especificación de requerimientos**

|  |  |
| --- | --- |
| **Título** | **Restablecer contraseña** |
| **Prioridad** | **Media** |
| **Diseño de la página** | **VISTA RESTABLECER CONTRASEÑA. UNA VEZ EL USUARIO DE CLIC EN CAMBIAR CONTRASEÑA SE DESPLEGARÁ LA VISTA CAMBIAR CONTRASEÑA**    **VISTA CAMBIAR CONTRASEÑA** |
| **Especificación de requerimientos** | Al presionar restablecer contraseña en el inicio de sesión, el sistema lo direcciona a esta página:  La página solicita que se indique si se desea restablecer contraseña por nombre de usuario o por correo.   * Restablecer contraseña por nombre de usuario:   Si se selecciona esta opción el sistema solicita:   * + Nombre   + Restablecer: Al presionar restablecer el sistema valida que exista el usuario, si no existe le dice que el usuario no existe y si existe le muestra un mensaje en pantalla indicando que será notificado vía e-mail. * Restablecer contraseña por correo:   Si se selecciona esta opción el sistema solicita:   * + Correo   + Restablecer: Al presionar restablecer el sistema valida que exista el usuario, si no existe le dice que el usuario no existe y si existe le muestra un mensaje en pantalla indicando que será notificado vía e-mail. |
| **Notificación restablecer contraseña**  El usuario recibirá una notificación a su correo electrónico que contendrá un enlace para restablecer su contraseña. Al hacer *click* en el enlace éste será redirigido a una página de cambiar contraseña. |
| **Cambiar contraseña:**  La página solicita:   * Contraseña: El usuario debe especificar una contraseña que tenga como mínimo 8 caracteres, entre mayúsculas, minúsculas y un carácter específico, si la contraseña no cumple con los requisitos mínimos el sistema debe enviar una advertencia de que la contraseña no cumple con los requerimientos. Campo requerido. * Cambiar: Botón para cambiar la contraseña, al presionar el sistema valida que la contraseña cumpla con los requisitos. Si cumple, se realiza el cambio, se muestra un mensaje de texto y se dirige a la pagina de inicio de sesión. Si no cumple se le muestra un mensaje indicando el contenido del campo. |
| **Criterios de aceptación** | * La pagina reestablecer tiene las opciones de cambiar por nombre o correo. * reestablecer por nombre solicita el nombre y habilita el botón cambiar. * reestablecer por correo solicita el correo y habilita el botón cambiar. * Al solicitar el restablecimiento se valida la existencia del usuario. * Se notifica el restablecimiento. * En la notificación hay un enlace que lleva a la página de cambiar contraseña. * En la pagina de cambiar contraseña de solicita la nueva contraseña. * Se valida la longitud y el formato del campo contraseña. * Al cambiar la contraseña el sistema informa el cambio y redirige a la pagina de inicio de sesión. |

**REQUERIMIENTO 4**

Ofrecer la opción para la gestión (CRUD) de imágenes para los usuarios autenticados. Es decir, el usuario puede cargar (crear) nuevas imágenes, y actualizar y/o dar de baja a imágenes existentes creadas por él. En la creación y/o actualización de una imagen el usuario deberá ingresar el nombre de la imagen y/o la imagen, y además definir si la imagen es privada o pública.

**Especificación de requerimientos**

|  |  |
| --- | --- |
| **Título** | **Crear imagen** |
| **Prioridad** | **Alta** |
| **Diseño de la página** |  |
| **Especificación de requerimientos** | En esta página se crearán las imágenes que usarán los usuarios autenticados. La página solicita:   * Nombre de la imagen - Campo de texto de máximo 30 caracteres. Permite caracteres alfanuméricos y que no permita caracteres especiales. Campo requerido. * Descripción de la imagen - Campo de texto de máximo 250 caracteres. Campo opcional. * Checkbox - Elegir si la imagen es pública o privada. Si la imagen es pública podrá ser visualizada por todas las personas que ingresen al sitio web. Si la imagen es privada solo puede ser accedida por el usuario que la subió. Campo requerido. * Botón adicionar foto - Botón que abrirá una ventana que permita cargar un archivo de tipo imagen (PNG, JPEG, JPG, BMP, GIF). Verifica que el tamaño de la foto sea menor al máximo requerido. Si no cumple las condiciones especificadas mostrar mensaje de advertencia “El formato no es adecuado” o “El tamaño de la fotografía supera el establecido”, según sea el caso. * Botón guardar - Botón que permite crear la imagen en el sistema si las validaciones son satisfactorias. Muestra mensaje de “publicación guardada con éxito”. |
| **Criterios de aceptación** | * Todos los campos ingresados por el usuario son guardados en el sistema. * Se verifica que el formato y tamaño de imagen cumplan con los requerido. * Se verifica que todos los campos requeridos estén diligenciados. |

|  |  |
| --- | --- |
| **Título** | **Editar imagen** |
| **Prioridad** | **Alta** |
| **Diseño de la página** | **VISTA EDITAR IMÁGENES. AL DAR CLIC SE DESPLIEGA LA VISTA EDITAR IMAGEN**  **VISTA EDITAR IMAGEN** |
| **Especificación de requerimientos** | Esta opción permite la edición de un registro de imagen ya existente de un usuario registrado en el sistema. Para esto el usuario dará clic sobre la imagen, lo cual mostrará la imagen en su tamaño original y debajo de la misma un botón “editar”, que al dar click sobre él se abrirá una página que permita modificar:   * Nombre de la imagen - Campo de texto de máximo 30 caracteres. Permite caracteres alfanuméricos y especiales. Campo requerido. * Descripción de la imagen - Campo de texto de máximo 250 caracteres. Campo opcional. * Checkbox - Elegir si la imagen es pública o privada. Si la imagen es pública podrá ser visualizada por todas las personas que ingresen al sitio web. Si la imagen es privada solo puede ser accedida por el usuario que la subió. Campo requerido. * Botón adicionar foto - Botón que abrirá una ventana que permita cargar un archivo de tipo imagen (PNG, JPEG, JPG, BMP, GIF). Verifica que el tamaño de la foto sea menor al máximo requerido. Si no cumple las condiciones especificadas mostrar mensaje de advertencia “El formato no es adecuado” o “El tamaño de la fotografía supera el establecido”, según sea el caso. * Botón guardar - Botón que permite actualizar la imagen creada en el sistema si las validaciones son satisfactorias. Muestra mensaje de “publicación guardada con éxito”. |
| **Criterios de aceptación** | * Todos los campos ingresados por el usuario son actualizados en el sistema. * Se verifica que el formato y tamaño de imagen cumplan con los requeridos. * Se verifica que todos los campos requeridos estén diligenciados. |

|  |  |
| --- | --- |
| **Título** | **Eliminar imagen** |
| **Prioridad** | **Media** |
| **Diseño de la página** | Incluido en la vista editar imagen |
| **Especificación de requerimientos** | Esta opción permite la eliminación de un registro de imagen ya existente de un usuario registrado en el sistema. Para esto el usuario dará clic sobre la imagen, lo cual mostrará la imagen en su tamaño original y debajo de la misma un botón “eliminar”. Al dar clic sobre él aparecerá una ventana pidiendo confirmación para la eliminación de la foto. La ventana tendrá dos botones:   * Botón eliminar - borra la imagen de la galería de imágenes. * Botón cancelar - cierra la ventana y vuelve a la vista de la imagen ampliada. |
| **Criterios de aceptación** | * Validar que la imagen quede eliminada al realizar la opción de eliminar por parte del usuario. |

|  |  |
| --- | --- |
| **Título** | **Visualizar imagen** |
| **Prioridad** | **Media** |
| **Diseño de la página** | **Incluido en las otras vistas** |
| **Especificación de requerimientos** | Funcionalidad que permite tener una vista previa de la imagen registrada. No se pueden modificar los datos de la imagen (nombre, descripción, privacidad).  Esta opción permite mostrar todas las imágenes que cumplan con ciertos criterios:   * Se mostrarán todas las imágenes que ha subido el usuario en su ventana principal una vez haya iniciado sesión. * En la página principal del sitio se mostrarán todas las imágenes que se encuentren en la base de datos que tengan el acceso como público. |
| **Criterios de aceptación** | * No se permite la edición de los datos de la imagen (nombre, descripción, privacidad). * Se muestra la imagen en un tamaño predeterminado. * En la página principal del sitio se muestran todas las fotos cargadas por todos los usuarios que posean acceso público. * Una vez el usuario inicie sesión puede visualizar todas las imágenes que ha publicado. |

**REQUERIMIENTO 5**

Buscar una imagen usando una palabra clave para el nombre. Esta búsqueda debe mostrar una galería de imágenes públicas que satisfacen la restricción.

**Especificación de requerimientos**

|  |  |
| --- | --- |
| **Título** | **Buscar imagen** |
| **Prioridad** | **Media** |
| **Diseño de la página** | **Dirigirse a Vista Inicial** |
| **Especificación de requerimientos** | La página de búsqueda debe incluir:   * Cuadro de texto vacío - Ubicado en la parte superior central donde se pueda ingresar una palabra clave para realizar la búsqueda. * Botón buscar - Botón que compara la palabra clave digitada con las cadenas de texto que se encuentran en el nombre de la imagen y en la descripción de la imagen. Si encuentra alguna coincidencia agrega esta imagen a los resultados a mostrar. Se debe verificar que la coincidencia sea una imagen pública. * Botón Limpiar - Borra lo que se haya escrito en el cuadro de texto de búsqueda. |
| **Criterios de aceptación** | * Se muestran las imágenes públicas que incluyan en el nombre o en la descripción la palabra clave. * Se muestran solo las imágenes que incluyan en el nombre de la imagen o en la descripción la palabra clave buscada. |

**REQUERIMIENTO 6**

Un usuario autenticado puede descargar una imagen pulsando un botón que debe estar disponible cuando la imagen es pública

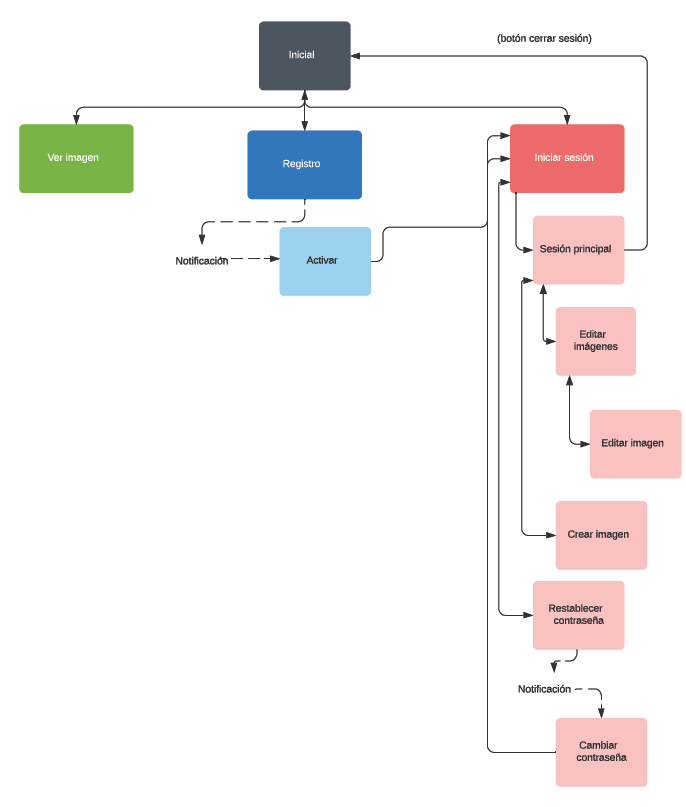
**Especificación de requerimientos**

|  |  |
| --- | --- |
| **Título** | **Descargar imagen** |
| **Prioridad** | **Media** |
| **Diseño de la página** | **Dirigirse a sesión Principal** |
| **Especificación de requerimientos** | Una vez el usuario de clic en una imagen, la imagen deberá ampliarse y tener en la parte inferior un botón “Descargar”. Este botón debe iniciar de manera automática la descarga del archivo en el formato en el que fue subida. |
| **Criterios de aceptación** | * Si un usuario no registrado trata de descargar una imagen debe mostrar un mensaje en el cual se le comunique al usuario que debe iniciar sesión para poder descargar la imagen. * Al dar clic en el botón “Descargar” la imagen se descarga automáticamente en el ordenador. |

# Diagrama de Clases

[](https://drive.google.com/file/d/1gyLcoM0dilRc6tAaG5LWRPPPiOva4RuF/view?usp=sharing)

# Navegabilidad



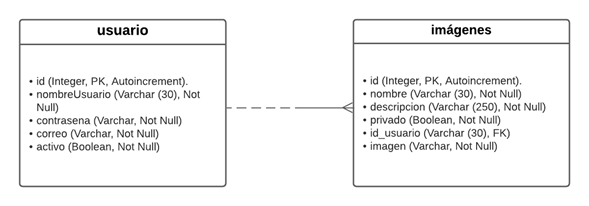
# Descripción funciones manejadoras

[**Archivo Funciones Manejadoras xls**](https://drive.google.com/file/d/1tIUePt-X8v3D9LsQQmskMHdWR9jHfoYo/view?usp=sharing)

# Implementaciones de seguridad y validaciones

|  |  |  |  |
| --- | --- | --- | --- |
| Función manejadora | Característica | Descripción | Captura del código |
| index | Seguridad: Prepared statements | Este query devuelve las imágenes cuyo nombre de imagen o descripción coincida con un texto ingresado por el usuario en la barra de búsqueda. | def sql\_buscar\_imagenes(texto\_busqueda):  db = get\_db()  strsql = "SELECT imagen FROM imagenes WHERE nombre LIKE ('%' || ? || '%') OR descripcion LIKE ('%' || ? || '%')"  resultado\_imagenes = db.execute(strsql,(texto\_busqueda,texto\_busqueda)).fetchall()  close\_db()  return resultado\_imagenes |
|  | Seguridad: Prepared statements | Este query devuelve las imágenes que cuya columna *privado* sea igual a *falso*. | def sql\_leer\_imagenes\_publicas():  db = get\_db()  strsql = 'SELECT imagen FROM imagenes WHERE privado = ?'  temp1 = "0"  imagenes\_publicas = db.execute(strsql,(temp1,)).fetchall()  close\_db()  return imagenes\_publicas |
| registroUsuario | Seguridad:  Prepared statements | Este query verifica si el correo que se va a registrar se encuentra previamente en la base de datos. | def sql\_id\_usuario\_por\_correo(correo):  db = get\_db()  strsql = 'SELECT id FROM usuario WHERE correo = ?;'  id = db.execute(strsql,(correo,)).fetchone()  return id |
|  | Seguridad: Prepared statements | Este query verifica si el nombre de usuario ya se encuentra registrado en la base de datos. | def sql\_id\_usuario(NombreUsuario):  db = get\_db()  strsql = 'SELECT id FROM usuario WHERE nombreUsuario = ?;'  id = db.execute(strsql,(NombreUsuario,)).fetchone()[0]  return id |
|  | Seguridad: Prepared statements | Este query ingresa los datos previamente validados a la base de datos | def sql\_nuevo\_usuario(nombreUsuario,correo,contrasena):  db = get\_db()  strsql = 'INSERT INTO usuario (nombreUsuario, correo, contrasena,activo) VALUES (?,?,?,?)'  db.execute(strsql,(nombreUsuario,correo,contrasena,"0"))  db.commit()  close\_db()  return "Usuario creado" |
|  | Seguridad:  Contraseña hash | Se encripta la contraseña con hash | hashContraseña = generate\_password\_hash(contrasena) |
|  | Validaciones:  Campos del formulario | Se verifica por medio de expresiones regulares que:  \* El nombre de usuario tenga entre 8 y 30 caracteres.  \* El correo sea de la forma [XXX@XXX.COM](mailto:XXX@XXX.COM)  \* La contraseña tenga mínimo 8 caracteres, que contenga una mayúscula, una minúscula y un número.  \* Que todos los campos estén diligenciados | pass\_reguex = "^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)[^\W\_]{8,}$"  user\_reguex = "^(?=.{8,20}$)"  email\_reguex = "^[^@]+@[^@]+\.[a-zA-Z]{2,}$"  usuario = TextField('Nombre de usuario', validators=[DataRequired(message="Campo requerido"),Regexp(user\_reguex,message ='El nombre de usuario debe tener mínimo 8 caracteres y máximo 20')],render\_kw={"placeholder": "miUsuario123", "class":"form-control","style":"text-transform: none"})  password = PasswordField('Password',validators=[DataRequired(message="Campo requerido"),Regexp(pass\_reguex,message='La contraseña debe contener mínimo 6 caracteres, un dígito, una mayúscula y una minúscula')],render\_kw={"placeholder": "Contraseña", "class":"form-control","style":"text-transform: none"})  email = EmailField('Correo electrónico',validators=[DataRequired(message="Campo requerido"),Regexp(email\_reguex,message ='El email debe ser de la forma micorreo@ejemplo.com')],render\_kw={"placeholder": "micorreo@example.com", "class":"form-control","style":"text-transform: none"}) |
|  | Validaciones con la base de datos | Se verifica que el usuario y el correo electrónico no estén registrados anteriormente en la base de datos | if sql\_id\_usuario\_por\_correo(correo) is not None:  error = 'El correo ya existe'.format( correo te("registroUsuario2.html",form = form)    elif sql\_)  flash ( error )  return render\_templaid\_usuario(nombreUsuario) is not None:  error2 = 'El nombre de usuario ya existe.'.format( nombreUsuario )  flash ( error2 )  return render\_template("registroUsuario2.html",form = form) |
| iniciarSesion | Seguridad: sesión | asignar a la variable global g.user el nombre de usuario para que la sesión pueda mantenerse mientras el usuario se encuentre loggeado | session.clear()  session['user\_id'] = user[0]  resp = make\_response( redirect( url\_for( 'inicialSesion' ) ) )  resp.set\_cookie('NombreUsuario', NombreUsuario)  load\_logged\_in\_user() |
|  | Seguridad: Contraseña | validar que la contraseña sea correcta | check\_password\_hash( user[1], contrasena ): |
|  | Seguridad: Prepared statements | Este query retorna los datos asociados a un nombre de usuario. Si no hay una coincidencia la función retornará None. | def sql\_iniciar\_sesion(nombreUsuario):  strsql = 'SELECT \* FROM usuario WHERE nombreUsuario = ?'  db = get\_db()  resultado = db.execute(strsql,(nombreUsuario,)).fetchone()  close\_db()  return resultado |
|  | Seguridad: Prepared statements | Este query retorna el estado de una cuenta: si está activa o no, a partir del nombre de usuario. | def sql\_usuario\_activo(nombreUsuario):  strsql = 'SELECT activo FROM usuario WHERE nombreUsuario = ?'  db = get\_db()  activo = db.execute(strsql,(nombreUsuario,)).fetchone()[0]  close\_db()  return activo |
|  | Validaciones: existencia de cuenta | Se valida si el usuario ingresado existe en la base de datos y si la cuenta se encuentra activa. | if user is not None:  activo = sql\_usuario\_activo(NombreUsuario)  if user is None:  error = 'Usuario o contraseña inválidos'    elif activo is None:  error = 'Usuario o contraseña inválidos'  elif activo is not None and activo == 0:  error = 'El usuario no esta activo' |
|  | Validaciones de campos ingresados | Se valida si los campos del formulario fueron diligenciados | iniciar = SubmitField('Iniciar',render\_kw={"class":"btn btn-defeault btn-send"})  usuario2 = TextField('Nombre de usuario', [DataRequired(message="Campo requerido")],render\_kw={"placeholder": "miUsuario123", "class":"form-control","style":"text-transform: none"})  password2 = PasswordField('Password',validators=[DataRequired(message="Campo requerido")],render\_kw={"placeholder": "Contraseña", "class":"form-control", "style":"text-transform: none"}) |
| activacionUsuario | Seguridad: Prepared statements | Este query envía una actualización al estado del campo de activación en la base de datos, asignando un 1 si el estado de activo es igual a 0, de no ser así devuelve un mensaje donde se le informa al usuario que ya se encuentra activo. | def sql\_activar\_usuario(nombreUsuario):  if sql\_usuario\_activo(nombreUsuario) == 0:  strsql = 'UPDATE usuario SET activo = 1 WHERE nombreUsuario = ?'  db = get\_db()  db.execute(strsql,(nombreUsuario,))  db.commit()  close\_db()  return flash("Cuenta activada")  elif sql\_usuario\_activo(nombreUsuario)== 1:  return flash("El usuario ya se encuentra activo. Por favor inicie sesión") |
|  | Validaciones con la base de datos | Este query se utiliza para realizar la lectura del campo activo y así poder activar al usuario o enviar mensaje de que ya se encuentra activo, este al igual que el query “sql\_activar\_usuario” están bajo la seguridad de Prepared statements. | def sql\_usuario\_activo(nombreUsuario):  strsql = 'SELECT activo FROM usuario WHERE nombreUsuario = ?'  db = get\_db()  activo = db.execute(strsql,(nombreUsuario,)).fetchone()[0]  close\_db()  return activo |
| inicialSesion | Seguridad: Prepared statements | Es este query se hace una extracción en la base de datos de imágenes que contengan palabras similares en el nombre o descripción que digita el usuario en la la vista inicial con sesion. | def sql\_buscar\_imagenes(texto\_busqueda):  db = get\_db()  strsql = "SELECT imagen FROM imagenes WHERE nombre LIKE ('%' || ? || '%') OR descripcion LIKE ('%' || ? || '%')"  resultado\_imagenes = db.execute(strsql,(texto\_busqueda,texto\_busqueda)).fetchall()  close\_db()  return resultado\_imagenes |
|  | Seguridad: Prepared statements | Es este query se realiza la petición a la base de datos para que me devuelva las imágenes donde la privacidad es igual a 0. | def sql\_leer\_imagenes\_publicas():  db = get\_db()  strsql = 'SELECT imagen FROM imagenes WHERE privado = ?'  temp1 = "0"  imagenes\_publicas = db.execute(strsql,(temp1,)).fetchall()  close\_db()  return imagenes\_publicas |
|  | Seguridad: Prepared statements | En este query primero se extrae el id del usuario con el nombre de usuario enviado desde la función y se extrae desde la base de datos la información de las imágenes, nombre, descripción y la imagen a partir del id del usuario. | def sql\_leer\_imagenes\_propias(nombreUsuario):  db = get\_db()  strsql = 'SELECT id FROM usuario WHERE nombreUsuario = ?'  id = db.execute(strsql,(nombreUsuario,)).fetchone()[0]  strsql2 = 'SELECT nombre, descripcion, imagen,id FROM imagenes WHERE id\_usuario = ?'  imagenes = db.execute(strsql2,(str(id),)).fetchall()  close\_db()  return imagenes |
|  | @login\_required y  extracción del id | En la función inicialSesion previamente se valida si la session del usuario está activa para poder ingresar, una vez hace esta validación se extrae el id del usuario con la variable global g.user[0] | @app.route("/inicialConSesion/", methods = ["GET", "POST"])  @login\_required #NO ESTA CREANDO EL USUARIO  def inicialSesion():  from\_id = g.user[0]  print("ID DEL G.USER = " + str(from\_id))  form = formBuscarImagen() |
| restablecerContrasena | Seguridad: Prepared statements | En este query se realiza la petición a la base de datos para validar si el correo que digitó el usuario se encuentra registrado, si es correcta la validación se hace el envío de un correo con un link para restablecer la contraseña. | def sql\_restablecer\_contrasena(correo):  strsql = 'SELECT correo FROM usuario WHERE correo = ? and activo= ?; '  db = get\_db()  resultado = 0  if db.execute(strsql,(correo,"1")).fetchone() is not None:  resultado = 1  close\_db()  return resultado |
|  | Envío de correo electrónico. | En esta sección se envía el correo a la persona con el link para hacer el cambio de contraseña. ( Por el momento en el link se envía una ruta local, la cual se modificará cuando ya se cree el dominio de la página) | yag = yagmail.SMTP('misiontic2022grupo11@gmail.com', '2022Grupo11') #modificar con tu informacion personal  yag.send(to=correo, subject='Restablecimiento de contraseña',  contents= ' Da click en el link para restablecer tu contraseña..'+" http://127.0.0.1:5000/cambiarContrasena/"+correo) |
| cambiarContrasena | Seguridad: Prepared statements | En este query se realiza el cambio de la contraseña en la base de datos por medio del correo electrónico registrado. | def sql\_cambiar\_contrasena(correo, contrasena):  strsql = 'UPDATE usuario SET contrasena = ? WHERE correo = ?;'  db = get\_db()  db.execute(strsql,(contrasena,correo))  db.commit()  close\_db()  return flash("Contraseña actualizada") |
|  | Seguridad:  Contraseña hash | Se encripta la contraseña con hash | @app.route("/cambiarContrasena/<string:correo>", methods = ["GET", "POST"])  def cambiarContrasena(correo):  form = formVerificarContrasena()  if request.method == 'POST':  contrasena=request.form ['password1']  hashContraseña = generate\_password\_hash(contrasena)  sql\_cambiar\_contrasena(correo,hashContraseña) |
| crearImagen | Seguridad: Prepared statements | Este query adiciona una imagen, junto con su descripción, nombre y privacidad a la base de datos. | def sql\_crear\_imagen(NombreUsuario,nombreImagen,descripcion,privado,imagen):  db = get\_db()  id\_usuario = sql\_id\_usuario(NombreUsuario)  strsql = 'INSERT INTO IMAGENES(nombre, descripcion, privado, id\_usuario, imagen) VALUES (?, ?, ?, ?, ?)'  db.execute(strsql,(nombreImagen,descripcion,privado,id\_usuario,imagen))  db.commit()  close\_db()  return print("Se creo con exito la imagen") |
|  | Seguridad: @login\_required y  extracción del id | Utiliza el decorador login\_required para mantener la sesión.  Si la sesión está activada entra al método sino, no lo hace. | @app.route("/crearImagen/", methods = ["GET", "POST"])  @login\_required  def crearImagen():  NombreUsuario = g.user[0] |
|  | Guardar imagen: request.files[‘file2’].save | Carga una imagen subida por el usuario y guarda la ruta en la base de datos. | imgSave = '.\static\images\\'+nombreImagen+nombreImg+'.jpg'  imgQuery = '..\static\images\\'+nombreImagen+nombreImg+'.jpg'  request.files['file2'].save(imgSave)  if privacidad == "Privada": estado = 1  sql\_crear\_imagen(NombreUsuario,nombreImagen,descripcion,estado,imgQuery) |
| editarImagen | Seguridad: Prepared statements | Actualiza los datos de una imagen para un usuario y un id de imagen. | def sql\_actualizar\_imagen(NombreUsuario,nombreImagen,descripcion,privado,imagen,idImagen):  db = get\_db()  id\_usuario = sql\_id\_usuario(NombreUsuario)  id\_imagen = idImagen  strsql= 'UPDATE IMAGENES SET NOMBRE = ?, DESCRIPCION = ?, PRIVADO = ?, ID\_USUARIO = ?, IMAGEN = ? WHERE ID = ?;'  db.execute(strsql,(nombreImagen,descripcion,privado,id\_usuario,imagen,id\_imagen))  db.commit()  close\_db()  return print("Se actualizo con exito la imagen") |
|  | Seguridad: Prepared statements | Carga la imagen que se va a editar (la imagen que se visualiza en la vista editarImagen) | def sql\_leer\_imagen\_propia(idImagen):  db = get\_db()  strsql = 'SELECT nombre, descripcion, imagen FROM imagenes WHERE id = ?'  imagen = db.execute(strsql,(idImagen,)).fetchone()  return imagen |
|  | @login\_required y  extracción del id | Utiliza el decorador login\_required para mantener la sesión. Si la sesión se encuentra activa ingresa al método editarImagen, se pasa por parámetro el Id de imagen. Se trae el nombre de usuario. | @app.route("/editarImagen/<string:idImagen>", methods = ["GET", "POST"])  @login\_required  def editarImagen(idImagen):  NombreUsuario = g.user[0] |
|  | Guardar imagen: request.files[‘file2’].save | carga una imagen subida por el usuario y guarda la ruta en la base de datos. | if request.files['file2'].filename == '':  imgQuery = imagen[2]  else:  imgSave = '.\static\images\\'+nombreImagen+nombreImg+'.jpg'  imgQuery = '..\static\images\\'+nombreImagen+nombreImg+'.jpg'  request.files['file2'].save(imgSave) |
| eliminarImagen | Seguridad: Prepared statements | Elimina una imagen en base de datos dado un id de imagen. | def sql\_eliminar\_imagen(idImagen):  db = get\_db()  strsql= 'DELETE FROM imagenes WHERE ID = ?;'  db.execute(strsql,(idImagen,))  db.commit()  close\_db() |
| editarImagenes | Seguridad: Prepared statements | Este query devuelve los datos relacionados con las imágenes que están asociadas a un nombre de usuario | def editarImagenes():  NombreUsuario = g.user[0]  imagen = sql\_leer\_imagenes\_propias(NombreUsuario)  return render\_template("vistaEditarImagenes.html",imagen=imagen,NombreUsuario = NombreUsuario) |

# Diagrama Relacional

Para asegurar que la información del usuario que accede a nuestra aplicación web InstaCreations sea persistente, se diseña una base de datos basándonos en un modelo Entidad-Relación como se muestra a continuación:

Se establecen dos tablas: usuario e imágenes, con una relación de uno a muchos. Para la **tabla usuario** se establece como “Primary Key” el campo ID con autoincremento. La t**abla imágenes,** en el campo id contará con un constraint de tipo “Primary Key”, a su vez tendrá id\_usuario como “Foreign Key”.

# Dirección del sitio

Dirección del sitio: <https://54.196.174.67/>

Nombre usuario: gabriela.franco

Contraseña: Prueba123

Nombre usuario: tinoeldaddy123

Contraseña: Prueba123