**Requisitos de Usuario**

| **Se le debe implementar al usuario el “CRUD” - (Alta)** | |
| --- | --- |
| Responsable= kevin y favio | |
| **Código** | **RF-10** |
| **Tipo** | Funcional |
| **Descripción** | formularios de inicio de sesión, registro, actualización de datos y botón de eliminar usuario. |
| **Entradas** | Todos los datos provenientes de los campos de los formularios y acciones de botones. |
| **Salidas** | En inicio de sesión cambia de pantalla a la pantalla principal de la aplicación, cambio de los formularios, mensaje de la creación de los usuarios, acciones, eventos o errores generales. |
|  | Responsable=manuel, roger |
| **Código** | **RB-11** |
| **Tipo** | Funcional |
| **Descripción** | Funciones para invocar las vistas de la base de datos que están relacionadas al CRUD del usuario. |
| **Entradas** | Petición de datos del usuario. |
| **Salidas** | Respuesta de la base de datos a la petición del usuario |
|  | Responsable=manuel |
| **Código** | **RD-12** |
| **Tipo** | Funcional |
| **Descripción** | Se debe tener una entidad para almacenar los datos correspondientes de los usuarios y despachar las vistas del CRUD. |
| **Entradas** | Toda o parcialmente la información del usuario. |
| **Salidas** | Respuesta del estado del CRUD o información del usuario. |

| **Recuperar contraseña - (Baja)** | |
| --- | --- |
| Responsable= kevin fabio | |
| **Código** | **RF-13** |
| **Tipo** | Funcional |
| **Descripción** | Generar formulario para recuperación de contraseña por correo,pedir pin enviado por correo |
| **Entradas** | Correo pin de verificación |
| **Salidas** | Generar nueva contraseña. |
|  | Responsable=manuel roger |
| **Código** | **RB-14** |
| **Tipo** | Funcional |
| **Descripción** | verificar correo , generar pin a correo ,pots de nueva contraseña |
| **Entradas** | Petición de datos del usuario. |
| **Salidas** | Respuesta de la base de datos a la petición del usuario. |
|  | Responsable=manuel |
| **Código** | **RD-15** |
| **Tipo** | Funcional |
| **Descripción** | Se debe tener una entidad para almacenar los datos correspondientes de los usuarios y despachar las vistas del CRUD. |
| **Entradas** | Toda o parcialmente la información del usuario. |
| **Salidas** | Respuesta del estado del CRUD o información del usuario. |

| **Sesión de usuario - (Baja)** | |
| --- | --- |
| Responsable= Kevin y favio | |
| **Código** | **RF-16** |
| **Tipo** | Funcional |
| **Descripción** | Generar inicio de sesión con botón de registro y recuperación. |
| **Entradas** | Nombre de usuario y contraseña o correo. |
| **Salidas** | Acceso a la pantalla principal de la aplicación o error de inicio de sección. |
|  | Responsable= Manuel y Roger |
| **Código** | **RB-17** |
| **Tipo** | Funcional |
| **Descripción** | Funciones para autenticar a los usuarios y gestionar el proceso de inicio de sesión. |
| **Entradas** | Credenciales de inicio de sesión |
| **Salidas** | Respuesta sobre el estado de inicio de sesión, si fue exitosa o un mensaje de error. |
|  | responsable=manuel |
| **Código** | **RD-18** |
| **Tipo** | Funcional |
| **Descripción** | Almacenamiento de información relacionada con la autenticación de usuarios, como registros de inicio de sesión y credenciales. |
| **Entradas** | Toda o parcialmente la información del usuario. |
| **Salidas** | Respuesta del estado del CRUD o información del usuario. |

**Requisitos de Interfaz**

| **Se le debe implementar al usuario el “CRUD” - (Alta) - (Alta)** | |
| --- | --- |
|  | |
| Responsable = kevin favio | |
| **Código** | **RF-1** |
| **Tipo** | Funcional |
| **Descripción** | Genera vista de adición de nueva tarea ,ventana flotante con formulario de nueva tarea ,nombre de tarea ,descripción de tarea, fecha límite |
| **Entradas** |  |
| **Salidas** | Vista Descripción de la nueva tarea |
|  |  |
| **Código** | **RB-1** |
| **Tipo** | Funcional |
| **Descripción** | Post de datos de almacenamiento a base de datos,realizar back tiempo de tarea , |
| **Entradas** | Datos de tarea |
| **Salidas** | Datos de tarea |
|  | Responsable =manuel |
| **Código** | **RD-1** |
| **Tipo** | Funcional |
| **Descripción** | Se debe tener una entidad para almacenar los datos correspondientes de los usuarios y despachar las vistas del CRUD. |
| **Entradas** | Toda o parcialmente la información del usuario. |
| **Salidas** | Respuesta del estado del “CRUD” o información del usuario. |

|  | |
| --- | --- |
|  | |
|  | |
| **Código** | **RF-1** |
| **Tipo** | Funcional |
| **Descripción** |  |
| **Entradas** |  |
| **Salidas** |  |
|  |  |
| **Código** | **RB-1** |
| **Tipo** | Funcional |
| **Descripción** |  |
| **Entradas** |  |
| **Salidas** |  |
|  |  |
| **Código** | **RD-1** |
| **Tipo** | Funcional |
| **Descripción** |  |
| **Entradas** |  |
| **Salidas** |  |

|  | |
| --- | --- |
|  | |
|  | |
| **Código** | **RF-1** |
| **Tipo** | Funcional |
| **Descripción** |  |
| **Entradas** |  |
| **Salidas** |  |
|  |  |
| **Código** | **RB-1** |
| **Tipo** | Funcional |
| **Descripción** |  |
| **Entradas** |  |
| **Salidas** |  |
|  |  |
| **Código** | **RD-1** |
| **Tipo** | Funcional |
| **Descripción** |  |
| **Entradas** |  |
| **Salidas** |  |

IDEAS

--- TodoList Aplicativo ---

1. Requisitos Funcionales y no Funcionales documento SRS.

2. Diseño UI y UX mockups bocetos de la aplicación.

3. Modelo de base de datos.

-- Requisitos Funcionales --

- Requisitos de interfaz?

El aplicativo debe permitir al usuario escribir en una campo el texto que corresponde

a la tarea que quiera hacer.

Organizar las tareas por grupo, etiquetar y categorizar dichas tareas por ese grupo.

Permitir al usuario asignar eventos que quiera colocar a la tarea en específico.

Recordatorio de las tareas, utilizar un software externo o api de calendario del sistema

para manejar fechas y establecer eventos de notificacion como recordatorio de la tarea

Establecer estados en la tarea que sea modificable por el usuario, los estados

pueden ser "hecho", "por hacer", "eliminado", "vencido", tambien son modificables

por la propia aplicación respecto a los eventos de este mismo.

Al eliminar las tareas disponer de una papelera donde pueda recuperar las tareas.

# Cuando no se encuentra la aplicación activa y recientemente estaba con dato insertados

# mantener con persistencia esos datos seria como un borrador (opcional)

Especificar información estadistica de la tarea metadados como, fecha creación, tiempo etc.

# Progreso del usuario.

- Requisitos de usuario

Se debe mantener las funcionalidades "CRUD" con las tareas del usuario: crear, leer, actualizar, eliminar

Disponer al usuario de un registro de datos para ser exclusivo y pueda ser identificado

en el aplicativo.

Disponer al usuario de un login con sus datos para crear una sección y vincular el dispositivo

y mantenerla activa en este.

Disponer al usuario de un apartado del logeo donde pueda recuperar la contraseña.

IDEAS hechas por fabio o requisitos directamente

Requerimientos funcionales

1.Agregar Tareas: Los usuarios pueden agregar nuevas tareas a su lista. Por ejemplo, si alguien necesita comprar en la tienda, se puede agregar una tarea llamada "Comprar en la tienda" a la lista.

2.Editar Tareas: Los usuarios pueden editar las tareas existentes. Esto significa que pueden cambiar el nombre de la tarea, la descripción, la fecha de vencimiento o cualquier otro detalle relacionado con la tarea.

3.Eliminar Tareas: Los usuarios pueden eliminar tareas de la lista una vez que las hayan completado o si ya no son relevantes.

4.Marcar Tareas como Completadas: Además de editar y eliminar, los usuarios pueden marcar las tareas como completadas cuando las hayan realizado. Esto permite llevar un registro de las tareas realizadas y las pendientes

5.Priorización: permiten a los usuarios asignar prioridades a las tareas, como alta, media o baja, para que sepa qué tareas son más urgentes.

6.Fecha de Vencimiento: Los usuarios pueden establecer fechas de vencimiento para sus tareas, lo que les ayuda a gestionar plazos y recordatorios

7.Categorización: los usuarios pueden organizar sus tareas en categorías o etiquetas para una mejor organización.

8.Notificaciones: enviar notificaciones o recordatorios a los usuarios cuando se acerca la fecha de vencimiento de una tarea.

9.Cambio de idioma(opcional): Permitir a los usuarios extranjeros tener una mejor experiencia

con la app

10. Límite de actividad(diario): Es posible que en el dia a dia de una persona pueda ocurrir

emergencia o por algún motivo no pueda hacer las tareas, por eso se agrega una función

para pasar esas tareas como prioridad para el dia siguiente en función del tiempo que

conlleva realizarlas.

11.Mapa semanal: esta función permite revisar las actividades que haz completado en el transcurso

de la semana, permitiendote ver la productividad y el panorama general de las actividades

que realizas en tu dia a dia.

12.Subtareas: esta función permite añadir subtareas dentro de las tareas, así como detalles básicos de la tarea, ubicaciones, entre otros

13.Personalización: permite agregar un modo oscuro o una lista de una paleta de colores diferentes

14.Compartir: esta función permite compartir tu lista de tareas con otros usuarios

15.Recordatorio: enviar una alarma que te notifique de las tareas que tienes pendientes y/o que son más urgentes

Requerimientos no funcionales:

Rendimiento:

Tiempo de carga: Establecer un tiempo de carga razonable para la aplicación.

Tiempo de respuesta: Definir un límite de tiempo aceptable para que la aplicación responda a las acciones del usuario.

Disponibilidad:

Disponibilidad del sistema: Especificar el porcentaje de tiempo que la aplicación debe estar disponible para los usuarios (por ejemplo, un 99.9% de disponibilidad).

Escalabilidad:

Escalabilidad: Definir cómo la aplicación manejará un aumento en la carga de trabajo y cómo se pueden agregar recursos para mantener el rendimiento.

Seguridad:

Autenticación y autorización: Especificar los requisitos de autenticación de usuarios y cómo se gestionarán los roles y permisos.

Protección contra ataques: Define las medidas de seguridad para proteger la aplicación contra amenazas como la inyección de SQL y los ataques de denegación de servicio.

Usabilidad:

Experiencia del usuario: Establecer estándares de usabilidad y realizar pruebas de usuario para garantizar una experiencia de usuario eficaz y satisfactoria.

Mantenibilidad:

Modularidad: Diseñar la aplicación de manera modular para facilitar futuras actualizaciones y cambios.

Documentación: Proporcionar documentación detallada para desarrolladores y administradores de sistemas.

Conformidad con Normativas:

Cumplimiento legal: Asegurar de que la aplicación cumple con todas las leyes y regulaciones aplicables, como las relacionadas con la privacidad de datos.

Eficiencia:

Uso de recursos: Define los requisitos de uso de recursos, como el consumo de CPU y memoria.

Tolerancia a Fallos:

Respuesta a fallos: Define cómo la aplicación manejará situaciones de fallos, como caídas del servidor o pérdida de conexión a la base de datos.