

Proyecto Final

Ingeniería de Requerimientos

Documentación del Proyecto

Participantes:

David Mata Guerra
Alejandro Macías Fonseca
Alexis Emilio Cárdenas Camacho
Gabriel Juárez Ramírez

Universidad Autónoma de Querétaro

8 de noviembre de 2024

Índice general

1. Descripción General del Proyecto	2
1.1. Resumen	2
1.2. Alcances del proyecto	2
1.3. Limitaciones del proyecto	2
1.4. Justificación	3
2. Especificación de Requerimientos	4
2.1. Requerimientos Funcionales	4
2.2. Stakeholders	4
2.3. Reglas de Negocio	4
2.4. Casos de Uso	5
2.5. Historias de Usuario	12
2.6. Requerimientos No Funcionales	12
3. Análisis y Diseño del Sistema	14
3.1. Frontend	14
3.2. Backend	14
3.2.1. Manejo de rutas y métodos	14
4. Planificación y Gestión del Proyecto	16
4.1. Metodología	16
4.2. Fuentes	16
4.3. Cronograma	16
4.4. Recursos y Roles	16
4.5. Control de Versiones	17
4.6. Herramientas de Desarrollo	17
5. Manual de Usuario	19
5.1. Guía de Uso	19
6. Documentación de Despliegue	20
6.1. Instrucciones de Instalación	20
7. Registro de Cambios (Changelog)	21

Capítulo 1

Descripción General del Proyecto

1.1. Resumen

El propósito de este proyecto es crear una herramienta de tareas, la cual pueda crear, editar, eliminar y marcar como completadas las tareas que el usuario desee. Además, se podrá visualizar las tareas que se han completado y las que están pendientes.

1.2. Alcances del proyecto

- Gestionar el proyecto mediante la herramienta JIRA utilizando la metodología Scrum.
- Desarrollar un prototipado con todas las funcionalidades que se dean implementar en Figma.
- Hacer uso un repositorio en GitHub para el control de versiones de el frontend.
- Hacer uso repositorio en GitHub para el control de versiones de el backend.
- Crear un diagrama relacional de la base de datos.
- Crear una base de datos con la estructura del diagrama relacional.
- Desarrollar la aplicación.
- Lanzar la aplicación en un servidor.
- Realizar test a la aplicación.
- Desglosar los pasos del proyecto en una presentación.

1.3. Limitaciones del proyecto

- El desarrollo de el proyecto tiene un tiempo limitado (3 semanas y 4 días).
- El proyecto no busca fines lucrativos.
- El proyecto se encuentra limitado a la creación de tareas.

1.4. Justificación

El proyecto surge en base a un proyecto final solicitado en la clase de Ingeniería de Requerimientos de la Universidad Autónoma de Querétaro. La idea de crear un gestor de tareas es con la finalidad de poner a prueba los conocimientos adquiridos en la materia y en la carrera.

Capítulo 2

Especificación de Requerimientos

2.1. Requerimientos Funcionales

- Gestión de tareas:
 - Crear tareas.
 - Editar tareas.
 - Eliminar tareas.
 - Marcar tareas como completadas.
- Visualización de tareas:
 - Interfaz intuitiva y práctica de utilizar.
 - El usuario pueda reconocer todos los cambios realizados de manera visual.

2.2. Stakeholders

- Desarrolladores.
- Usuarios.
- Profesor.

2.3. Reglas de Negocio

RN1. Las tareas deben de pertenecer a una persona

RN2. Las tareas deben de tener un nombre

RN3. Las tareas hija no deben de pertenecer a múltiples tareas padre

RN4. Las tareas pueden tener una descripción

RN5. Las tareas pueden tener una fecha

RN6. Las tareas tienen dos estados

- Sin completar
- Completado

RN7. Las tareas pueden tener 4 niveles de prioridades.

- Ninguna
- Baja
- Media
- Alta

El valor predeterminado es “Ninguno”

RN8. Las tareas pueden ser recurrentes

2.4. Casos de Uso

Id y nombre:	CU-1. Entrar a la plataforma		
Creado por:	Alejandro		
Actor principal:	Usuario final	Actores secundarios:	Base de datos, aplicación principal
Descripción	El usuario final entrará a la plataforma dando un correo electrónico y una contraseña		
Trigger:	El usuario final al entrar a la <i>landing page</i> , tendrá la opción de entrar a la plataforma y utilizarla		
Precondiciones:	<ul style="list-style-type: none"> • PRE-1. El usuario final cuenta con conexión a internet • PRE-2. El usuario final cuenta con un correo electrónico 		
Postcondiciones:	<ul style="list-style-type: none"> • POST-1. El usuario tendrá acceso a sus tareas 		
Flujo normal:	1.0. Entrar a la plataforma <ul style="list-style-type: none"> • El usuario desde la <i>landing page</i> interactúa con el botón “Entrar” • La aplicación redirige al usuario final a una pantalla con el respectivo inicio de sesión • El usuario ingresa su correo y contraseña en los apartados • Una vez ingresados los dos campos, el usuario interactúa con el botón “Ingresar”, y estos datos son enviados • El sistema valida en la base de datos que el usuario esté registrado • Una vez confirmados los datos, el usuario será redirigido a la pantalla principal donde verá sus tareas 		
	1.1. Registro de usuario nuevo <ul style="list-style-type: none"> • En caso de que el usuario no esté registrado, tendrá la opción de registrarse mediante un botón con el que interactuará • La aplicación redirigirá al usuario a una pantalla donde ingresará su nombre, correo electrónico y contraseña 		
Flujo alternativo:			

		<ul style="list-style-type: none"> • El sistema validará la petición • Una vez validada la petición, el usuario tendrá acceso a la plataforma
Excepciones:		<hr/> <p>1.0.E1. Correo o contraseña incorrectos</p> <ul style="list-style-type: none"> • Si la validación no encuentra al usuario con el correo o contraseña especificados, el sistema devuelve un error • La aplicación atrapa el error, y muestra al usuario que el usuario o contraseña son incorrectos <p>1.1.E2. Usuario ya registrado</p> <ul style="list-style-type: none"> • Si el sistema al validar el registro del nuevo usuario, encuentra a uno con ese correo electrónico ya registrado, arroja un error • La aplicación atrapa este error y muestra que el usuario ya ha sido registrado
Prioridad		Alta
Frecuencia de uso	de	Al menos una vez por semana
Reglas de negocio		RN1.

ID y nombre	CU-2. Crear una tarea		
Creado por:	Alejandro	Fecha de creación	24-11-05
Actor principal:	Usuario final	Actores secundarios:	Base de datos, aplicación principal
Descripción	El usuario final especificará una tarea a crear con el botón para crear una nueva tarea. Tendrá que especificar el nombre de la tarea (obligatorio), una descripción (opcional) y una fecha de la tarea (opcional)		
Trigger:	El usuario final indica que quiere crear una tarea con el botón especificado		
Precondiciones:	<ul style="list-style-type: none"> • PRE-1. El usuario está registrado en la base de datos • PRE-2. El usuario cuenta con conexión a internet para acceder al sistema 		
Postcondiciones:	<ul style="list-style-type: none"> • POST-1. La tarea será guardada en la base de datos • POST-2. La tarea será mostrada en la página principal del usuario 		
Flujo normal:	2.0. Crear una tarea <ul style="list-style-type: none"> • El usuario final pide al sistema crear una nueva tarea mediante la aplicación • La aplicación despliega los campos a llenar para la tarea • Una vez llenados los campos, el usuario interactúa con la aplicación para guardar la tarea • La tarea es enviada a la base de datos del sistema y la valida 		
Flujo alternativo:	2.1. Crear una tarea anidada <ul style="list-style-type: none"> • Al seleccionar una tarea, la aplicación le mostrará al usuario distintos campos para editar, en uno de ellos, el agregar tareas anidadas • Al seleccionar el campo, se desplegará un campo donde el usuario dará el nombre de la tarea, junto con una opción para guardar la tarea • Al indicarle el usuario a la aplicación que quiere guardar la tarea anidada, esta enviará al sistema que se quiere crear una tarea anidada, pasándole el identificador de la tarea anidada, junto con el identificador de la tarea padre como campo • El sistema guarda la tarea anidada en la base de datos 		
Excepciones:	2.0.E1. La tarea ya ha sido creada <ul style="list-style-type: none"> • Si la validación de la base de datos encuentra que la tarea ya ha sido creada, arroja un error • La aplicación escucha si la base de datos arrojó el error, y muestra al usuario que la tarea ya ha sido creada • El usuario no podrá guardar la tarea hasta que el campo del título sea cambiado 		
Prioridad:	Alta		

Frecuencia de uso:	Aproximadamente de 1 a 5 veces por día por usuario
Reglas de negocio:	RN2., RN3., RN4., RN5., RN6., RN7.
Información adicional:	La aplicación deberá de mostrar en una ventana emergente los campos a llenar
Asumciones:	Las tareas subidas no se tomarán como completadas

ID y nombre	CU-3. Eliminar una tarea		
Creado por:	Alejandro	Fecha de creación	24-11-05
Actor principal:	Usuario final	Actores secundarios:	Base de datos, aplicación principal
Descripción	El usuario eliminará de sus tareas una tarea especificada por este por medio de la aplicación.		
Trigger:	El usuario final indica que quiere eliminar una tarea con el botón especificado		
Precondiciones:	PRE-1. El usuario está registrado en la base de datos PRE-2. El usuario cuenta con conexión a internet para acceder al sistema PRE-3. El usuario cuenta con al menos una tarea registrada en la base de datos		
Postcondiciones:	POST-1. La tarea será eliminada permanentemente en la base de datos POST-2. La tarea dejará de ser mostrada mostrada en la página principal del usuario		
Flujo normal:	3.0. Eliminar una tarea <ul style="list-style-type: none"> • El usuario final pide al sistema eliminar una tarea seleccionada mediante la aplicación • La aplicación despliega una ventana emergente con un mensaje de que quiere corroborar eliminar la tarea, junto con dos opciones, las cuales son “Cancelar” y “Confirmar”. En caso de seleccionar “Cancelar”, el sistema no borra la tarea y la ventana emergente desaparece • Una vez confirmada la elección, la aplicación envía la petición a la base de datos de eliminar la tarea seleccionada • La tarea es eliminada de la base del datos del sistema 		
Flujo alternativo:	3.1. Eliminar múltiples tareas <ul style="list-style-type: none"> • El usuario mediante la opción de seleccionar múltiples tareas que provee la aplicación, escoge distintas tareas destinadas a eliminar • El usuario utiliza la implementación de eliminar tarea que provee la aplicación para eliminar las tareas seleccionadas previamente • La aplicación envía al sistema los identificadores de las tareas a eliminar • El sistema hace la petición a la base de datos para eliminar los registros de las tareas especificadas • Las tareas son eliminadas de la base de datos 		
Prioridad:	Alta		
Frecuencia de uso:	Aproximadamente de 1 a 2 veces por día por usuario		

Reglas de negocio:	Por definir
Información adicional:	La aplicación deberá de mostrar en una ventana emergente el mensaje de confirmación
Asumciones:	El usuario tiene al menos una tarea registrada

ID y nombre	CU-4. Editar una tarea		
Creado por:	Alejandro	Fecha de creación	24-11-06
Actor principal:	Usuario final	Actores secundarios:	Base de datos, aplicación principal
Descripción	El usuario podrá editar una tarea por medio de la aplicación.		
Trigger:	El usuario final indica que quiere editar una tarea al entrar en una tarea		
Precondiciones:	PRE-1. El usuario está registrado en la base de datos PRE-2. El usuario cuenta con conexión a internet para acceder al sistema PRE-3. El usuario cuenta con al menos una tarea registrada en la base de datos		
Postcondiciones:	POST-1. La tarea será editada en la base de datos POST-2. La tarea será mostrada con su nueva información en la página principal del usuario		
Flujo normal:	4.0. Editar una tarea <ul style="list-style-type: none"> • El usuario final selecciona una tarea • La aplicación despliega una ventana con los datos de la tarea, dejándolos abiertos para editarlos • Una vez el usuario haya editado la tarea a voluntad, el usuario interactúa con la aplicación en un botón dedicado para guardar la tarea • La aplicación envía al sistema que se quieren actualizar los datos de la tarea especificada • El sistema valida la actualización • El sistema actualiza la información de la tarea especificada en la base de datos 		
Excepciones	4.0.E1. Renombrar como una tarea ya creada <ul style="list-style-type: none"> • En caso de que el nombre de la tarea ya exista, el sistema arroja un error de que la tarea ya existe. • La aplicación atrapa el error, y despliega en una ventana emergente el error. 		
Prioridad:	Alta		
Frecuencia de uso:	Aproximadamente de 1 a 2 veces por día por usuario		
Reglas de negocio:	Por definir		
Asumciones:	El usuario tiene al menos una tarea registrada		

ID y nombre:	CU-5. Visualizar en orden específico las tareas		
Creado por:	Alejandro		
Actor principal:	Usuario final	Actores secundarios	Base de datos, aplicación principal
Descripción:	El usuario podrá visualizar en órdenes predefinidos sus tareas, complementando la visualización predeterminada de estas		
Trigger:	El usuario final interactúa con la aplicación para agregar una jerarquía de visualización a la existente por prioridades		
Precondiciones:	<ul style="list-style-type: none"> • PRE-1. El usuario cuenta con conexión a internet • PRE-2. El usuario está registrado en la base de datos • PRE-3. El usuario tiene al menos una tarea creada 		
Postcondiciones:	<ul style="list-style-type: none"> • POST-1. El estado de ordenamiento será guardado si se vuelve a visitar el sitio • POST-2. La visualización no afectará al contenido de las tareas 		
Flujo normal:	5.0. Visualizar en orden específico las tareas <ul style="list-style-type: none"> • El usuario indica desde el apartado el nuevo orden de visualización • La aplicación recibe la petición, y esta reordena los elementos, sin dejar de respetar los apartados de prioridad 		
Prioridad:	Media		
Frecuencia de uso:	Aproximadamente una vez por sesión por usuario		

2.5. Historias de Usuario

- HU1. Como usuario, quiero poder crear nuevas tareas para organizar y gestionar mis actividades diarias.
- HU2. Como usuario, quiero poder ver una lista de todas mis tareas pendientes para tener una visión clara de lo que necesito hacer.
- HU3. Como usuario, quiero poder marcar las tareas como completadas para llevar un seguimiento de mi progreso.
- HU4. Como usuario, quiero poder establecer fechas límite para mis tareas para asegurarme de cumplir con mis plazos.
- HU5. Como usuario, quiero poder editar mis tareas para poder hacer cambios en ellas si es necesario.

2.6. Requerimientos No Funcionales

- Rendimiento:

- No debe de haber errores en la logica de programacion.
- La aplicación debe de ser rápida y eficiente.

Capítulo 3

Análisis y Diseño del Sistema

3.1. Frontend

3.2. Backend

3.2.1. Manejo de rutas y métodos

- `'/'`.

En la ruta principal, el usuario entrará a la *landing page*, donde se mostrarán los objetivos básicos de la aplicación, y el apartado respectivo para entrar a la aplicación

- `'/login'`.

- GET. Se desplegará la pantalla para iniciar sesión, donde estarán los campos competentes para iniciar sesión.
- POST. Se enviarán las credenciales del usuario que quiera iniciar sesión.

- `'/app/:idUser/tasks'`

- GET. Se mostrarán todas las tareas que el usuario tenga (mostrará como tareas principales aquellas que no tengan definido su parámetro de tarea padre).
- POST. La ruta recibirá en el formato correspondiente las tareas a crear

- `'/app/:idUser/tasks/:idTask'`

- GET. Al acceder a esta ruta, se recibirá la información de la tarea pasada por `idTask`
- PUT. Se actualizará la información de la tarea correspondiente.
- DELETE. Se eliminará la tarea especificada

- `'/app/:idUser/tasks/:idTask/subtasks'`

- GET. Se obtendrán todas las tareas hijas de la tarea pasada por `idTask`
- POST. Se añadirá la tarea hija correspondiente a la tarea pasada por `idTask`

- `'/app/:idUser/tasks/:idTask/subtasks/:idSubtask'`

- GET. Obtener una tarea hija específica
- PUT. Actualiza los parámetros de la tarea hija especificada.
- DELETE. Elimina una tarea hija especificada

Capítulo 4

Planificación y Gestión del Proyecto

4.1. Metodología

Se utilizó la metodología SCRUM, ya que con esta se tienen todos los requerimientos a la mano, donde cada *issue* tiene que ser definido detalladamente, sea una tarea, una historia de usuario, o un bug. Esta metodología ata los *issues* a un bloque de tiempo determinado llamado *sprint*, donde así, las fechas de entrega pueden definirse de una manera más clara, y el manejo del tiempo de los desarrolladores es gestionado de mejor manera.

4.2. Fuentes

- Repositorio Backend de GitHub
- Repositorio Frontend de GitHub
- JIRA

4.3. Cronograma

4.4. Recursos y Roles

Recursos:

- 1 Líder de Proyecto
- 2 Desarrolladores

Roles:

- Líder de Proyecto: Alejandro Macías Fonseca
- Desarrollador Backend: Alejandro Macías Fonseca
- Desarrollador Backend: David Mata Guerra
- Desarrollador Frontend: Alexis Emilio Cárdenas Camacho
- Desarrollador Frontend: Gabriel Juárez Ramírez

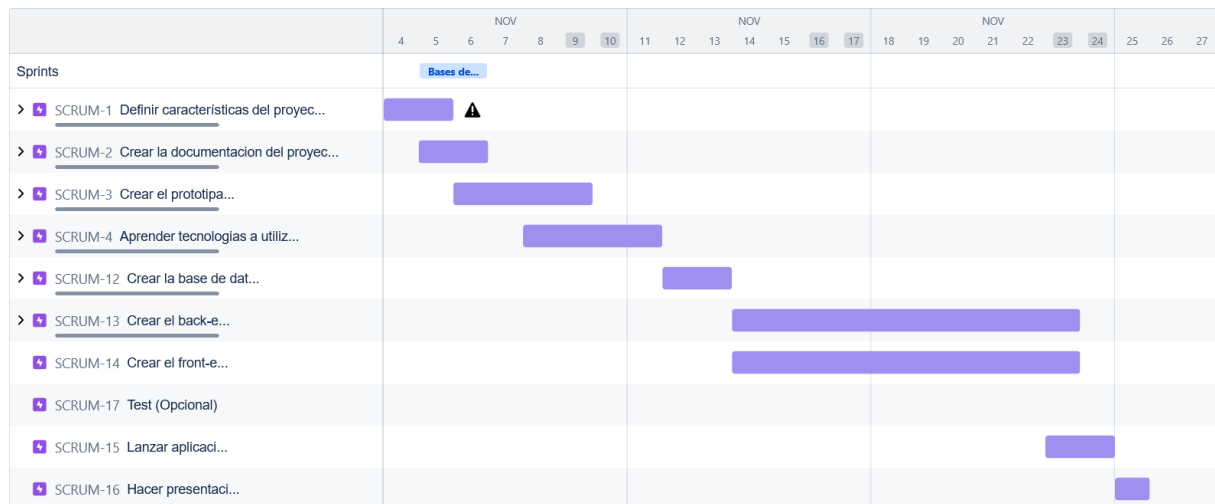


Figura 4.1: Cronograma del Proyecto obtenido de JIRA

4.5. Control de Versiones

El control de versiones se llevó a cabo utilizando Git y GitHub.



Se crearon dos repositorios :

- Backend: [Repositorio de GitHub](#)
- Frontend: [PORDEFINIR](#)

4.6. Herramientas de Desarrollo

- Documentación:
 - LaTeX
- Backend:
 - Node.js
 - Nest.js
 - PostgreSQL
 - Prisma

- Frontend:
 - React
 - Next.js
 - Shadcn

Capítulo 5

Manual de Usuario

5.1. Guía de Uso

Instrucciones para el usuario final.

Capítulo 6

Documentación de Despliegue

6.1. Instrucciones de Instalación

Cómo instalar y desplegar el sistema.

Capítulo 7

Registro de Cambios (Changelog)

Documente los cambios y versiones del sistema.