

Estructura de formularios y tablas

Cliente	Lunis Ing.	Proyecto	BerryTask			Version 1.0	Rev 1.2
Fecha	Jul 01, 2023	Ultima Act.	AndresL	Modifico	AndresL	Fecha mod.	8/12/2023

Descripción general de la aplicación

BerryTask (Nombre clave del proyecto) deberá ser una aplicación web cuyo objetivo es la gestión de flujos de trabajo, organización de equipos, proyectos y gestión de servicio a clientes.

Objetivos específicos.

1. Crear y gestionar grupos o equipos de trabajo.
2. Crear proyectos de trabajo a donde se puedan agregar tareas asignadas a miembros de equipo de trabajo.
3. Crear tareas y subtareas en forma de árbol, donde una subtaska puede a su vez, tener más subtareas.
4. Crear tareas recurrentes.
5. Asignar tareas a usuarios específicos.
6. Notificar cuando una tarea se ha cumplido.
7. Asignar a las tareas una fecha objetivo, la cual representa el límite de tiempo que se tiene para cumplir con la tarea asignada.
8. Calcular el cumplimiento de tareas por usuarios, tareas y proyectos.
9. Realizar comentarios en las tareas.
10. Adjuntar archivos e imágenes en las tareas
11. Crear tickets de seguimiento para casos específicos o solicitudes específicas para realizar una actividad y que tengan un seguimiento final.
12. Mantener un control de importancia para cada tarea.

1. 1. Formularios y tablas

Tabla	Descripción
users	Es la tabla con los usuarios de la aplicación y los accesos. Utilizar Tabla usuarios existente en los proyectos actuales
projects	Tabla para gestionar proyectos, un proyecto estará relacionado con múltiples tareas y con múltiples usuarios, para que un usuario pueda participar en un proyecto deberá ser asignado por un administrador al proyecto, si un usuario no este asignado al proyecto no podrá ver la información ni el proyecto.
tasks	Toda la información de la tarea, una tarea podría estar relacionada con una tarea padre y así sucesivamente.
priorities	Catalogo de prioridades usadas en las tareas
comments	Almacena todos los comentarios del sistema

category_task	categoría de las tareas, es un catálogo.
customers	Cientes para asignar tareas
files	Archivos y recursos adjuntos en las tareas, están relacionadas a clientes, tareas y proyectos.
Category_files	Tipos de archivos, es un catalogo

1.1. Tabla *projects*

La estructura es la siguiente:

Nombre del campo	Tipo	Descripción
id	int	ID del registro, autoincrementadle
uuid	VARCHAR (25)	Es el identificador único del registro que garantiza en una muy baja probabilidad que el registro se repita
name	VARCHAR	Es el nombre del proyecto
Description	Short text	Es una breve descripción del proyecto
Members	Text	Miembros que pertenecen al proyecto, son los ID de usuario que tienen acceso al proyecto.
Date_start	Fecha	Fecha de inicio del proyecto
Date_end	Fecha	Fecha de finalización del proyecto
Status	Int	Estados del proyecto, Nuevo, En proceso, En riesgo, Retrasado, Terminado
Customer_id	Int	ID del cliente que podría estar relacionado al proyecto, no es un campo requerido.
Customer_name	VARCHAR	Nombre del cliente
creado_por	INT	Id de usuario que creo el registro
creado_fecha	DATETIME	
ultima_mod_por	INT	Id de usuario que modifiko el registro
ultima_mod_fecha	DATETIME	
modificaciones	TEXT	

1.2. Tabla *tasks*

La tabla *tareas* almacena la información de las tareas creadas por los miembros del grupo de trabajo, una tarea puede tener múltiples tareas hijo y a su vez cada tarea hijo puede tener múltiples tareas hijo.

La estructura es la siguiente:

Nombre del campo	Tipo	Descripción
Id	Int	Es el identificador único del registro y tiene un valor auto incrementable.
uuid	VARCHAR (25)	Es el identificador único del registro que garantiza en una muy baja probabilidad que el registro se repita
Parent_task_id	Int	En el caso de que la tarea sea una tarea hijo, este campo debería contener el ID de la tarea padre para relacionarla.
Project_id	Int	ID del proyecto al que pertenece la tarea, por omisión una tarea siempre deberá tener un proyecto asignado.
Task_name	VARCHAR	Es el nombre asignado a la tarea, es una descripción general
Description	Text	Es una descripción específica de la tarea, sería muy importante ver si en este campo podemos realizar una edición de texto más completa, usar tipografías, saltos, numeraciones, etc. Para permitir que el usuario pueda describir más exacto la idea de la tarea.
Due_date	Fecha	Es la fecha en que deberá ser terminada la tarea, si no se cumple esa tarea cambiará su estado.
Responsable	INT	ID del usuario que tiene que realizar la tarea.
Creador	INT	ID del usuario que creo la tarea.
customer_id	Int	ID del cliente al que pertenece la tarea, en caso de estar relacionado el proyecto con un cliente.
Categoría		categoría a la que pertenece la tarea
Date_start	DateTime	Fecha y hora en que debe de iniciar la tarea (Selecciona el usuario)
Date_end	DateTime	Fecha y hora en que se terminó la tarea (La guarda automáticamente)
Due_time	TIME	Hora a la que se tiene que ejecutar la tarea.
status	INT	Estado actual de la tarea, Nueva, En proceso, En riesgo, Retrasada, Finalizada
creado_fecha	INT	Fecha en que se creó la tarea
creado_por	INT	Usuario que creo la tarea

1.3. Tabla *priorities*

La tabla *priorities* almacena la información de cada una de las prioridades que pueden ser usadas en la aplicación y gestiona un color para cada prioridad.

La estructura es la siguiente:

Nombre	Tipo	Descripción
id	Int	Id del registro en la base de datos, auto incrementable
name	VARCHAR	Es el nombre para mostrar para las prioridades, Ejemplo: Baja, Media, Alta, Critico.
color	VARCHAR	Color en HEX con el que se identifica la prioridad y con esa deberá pintarse en los formularios.
creado_por	INT	Id de usuario que creo el registro
creado_fecha	DATETIME	Fecha que se creó el registro
ultima_mod_por	INT	

1.4. Tabla *comments*

La tabla *comments* deberá generar un historial de comentarios realizados por usuarios colaboradores del grupo de trabajo sobre la tarea o proyecto en cuestión.

Nombre	Tipo	Descripción
id	Int	Id del registro en la base de datos, auto incrementable
user	int	ID de usuario que comento
Id_task	Int	Id de la tarea donde se comento
Uuid	Varchar	UUID del comentario, se genera automáticamente
Created_date_time	datetime	Fecha y hora que se creó el comentario
Comment	Text	Mensaje de texto que se escribió o el comentario

1.5. Tabla *category_task*

La tabla *category_task* es una clasificación de las tareas asignadas a los proyectos, de tal forma que toda tarea debería tener siempre asignada una categoría.

Nombre	Tipo	Descripción
id	Int	Id del registro en la base de datos, auto incrementable
name	int	Nombre de la categoría, es como se muestra en los formularios
Created_date_time	datetime	Fecha y hora que se creó el comentario
Color	Varchar	Es el valor en HEX del color con el que se debe de pintar la categoría en los fomularios.

1.6. Tabla *customers*

La tabla customers almacena todos los clientes que pueden ser relacionados en los procesos de la aplicación. Una tarea podría o no estar relacionada con un cliente, un proyecto podría estar relacionado con un cliente y si un proyecto ya tiene asignado un cliente todas las tareas relacionadas a ese proyecto deberán estar relacionadas con ese cliente.

Nombre	Tipo	Descripción
id	Int	Id del registro en la base de datos, auto incrementable
name	int	Nombre de la categoría, es como se muestra en los formularios
address	VARCHAR	dirección del cliente
Phone	VARCHAR	Número de teléfono del cliente
Email1	VARCHAR	dirección de correo electrónico del cliente.
Email2	VARCHAR	dirección de correo electrónico del cliente.
Email3	VARCHAR	dirección de correo electrónico del cliente.
Email4	VARCHAR	dirección de correo electrónico del cliente.
Contact	VARCHAR	Nombre del contacto del cliente
Created_date_time	datetime	Fecha y hora que se creó el comentario

1.7. Tabla *files*

La tabla files almacena todos los archivos y recursos proporcionados por los clientes como imágenes, documentos, archivos de texto, pdf, etc.

Nombre	Tipo	Descripción
id	Int	Id del registro en la base de datos, auto incrementable
name	int	Nombre del recurso
Uuid	VARCHAR	UUID para identificar el recurso, este uuid deberá de ser el nombre con el que almacenamos el archivo dentro del directorio de archivos de la aplicación.
Type	VARCHAR	Es el tipo de archivo, clasificado tomando como base la extensión de los archivos, Imágenes (JPG, JPEG, BMP, GIF, PNG, ETC) PDF (PDF) Hojas de calculo (XLS), Documentos de Word (DOC, DOCX) Etc.
Create_date_time	VARCHAR	dirección de correo electrónico del cliente.
Expiration_date	DATETIME	Podría ser que el documento tiene fecha de expiración, aquí especificar para que se pueda dar seguimiento.
Files_category	INT	categoría para clasificar los archivos según su uso.
Email4	VARCHAR	dirección de correo electrónico del cliente.
Contact	VARCHAR	Nombre del contacto del cliente
Created_date_time	datetime	Fecha y hora que se creó el comentario

1.8. Tabla *category_files*

La tabla category_files es una clasificación de los documentos que pueden estar agregados en las tareas y proyectos, el objetivo es tener un repositorio de toda la información que se agregue en las tareas.

Nombre	Tipo	Descripción
id	Int	Id del registro en la base de datos, auto incrementable
name	int	Nombre de la categoría, es como se muestra en los formularios
Created_date_time	datetime	Fecha y hora que se creó el comentario
Color	Varchar	Es el valor en HEX del color con el que se debe de pintar la categoría en los formularios.

2. Requerimientos y alcances específicos.

Desarrollar una aplicación web responsive, intuitiva y atractiva para el usuario final donde se puedan ir creando proyectos y tareas diarias que deberán cumplir los usuarios o miembros de los grupos de trabajo.

Requerimientos específicos que deberían cumplirse para las tareas.

- Usar icono que permita señalar la tarea como finalizada y las tareas finalizadas deberían pintarse con un color diferente en el icono.
- Mostrar un icono que nos indique que la tarea tiene información adicional, al abrirla mostrara todo el contenido de la tarea.
- Mostrar el o los usuarios asignados a la tarea.
- Mostrar su categoría
- Mostrar fecha de entrega
- Una tarea deberá poderse copiar para generar una nueva tarea con toda la información relacionada a la tarea original y deberá incluir subtareas relacionadas, pero a la copia siempre deberemos poner en el nombre la palabra *copy* al final del nombre original. Por ejemplo, si la tarea original se llama “Llamar a todos los clientes” la copia debería llamarse “llamar a todos los clientes copy” y el usuario reescribirá el nombre que desee para la nueva tarea. Las fechas, categorías, miembros del grupo de trabajo y todas las subtareas en la tarea original deberán copiarse a la nueva tarea y conservar sus nombres originales.
- La aplicación se gestionará en tres paneles para navegar, uno con todos los posibles accesos a diferentes fuentes de tareas, Proyectos, Clientes, Mis tareas, Catálogos, Tickets (los tickets son una tarea con una clasificación que se configura en categorías), El panel central mostrara todas las tareas asignadas dependientes del proyecto y el panel de la derecha mostrara la información de la tarea seleccionada. Las subtareas se pueden mostrar debajo de la tarea principal.

Ejemplo:

3. Modulo

3.1 Proyectos

Lista: Listar todos los proyectos en forma tradicional como lo hacemos con otras listas para no batallar. Es decir, generar una lista de la forma tradicional que lo hacemos. La información que necesitamos en la lista es la siguiente:

1. Numero del proyecto: Es el ID del proyecto
2. Nombre del proyecto
3. Cliente: Si existe, si no vacío
4. Fecha de inicio
5. Fecha de entrega
6. Las Iniciales de los miembros
7. Progreso: En el progreso vamos a poner el % de avance del proyecto que se calcula de la siguiente manera:

$$\text{Progreso} = \text{Tareas completadas} / \text{Total de tareas del proyecto} * 100\%$$

8. Estado: Estado actual del proyecto, En proceso, Completo, Cancelado.

Sugerencias:

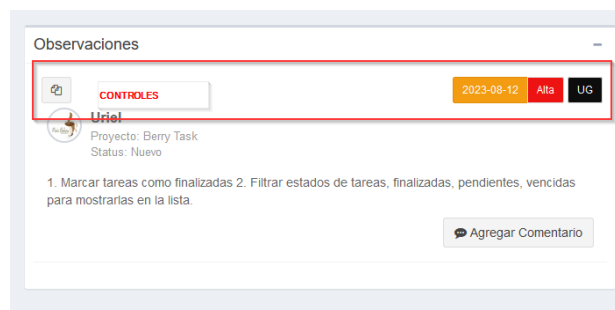
Condensed Full Width Table		
#	Task	Progress
1.	Update software	55%
2.	Clean database	70%
3.	Cron job running	30%
4.	Fix and squish bugs	90%

Nota: Al hacer clic en un proyecto debemos mostrar todas las tareas relacionadas a ese proyecto.

3.2 Tareas

Observaciones:

1. Acciones. En el panel de la tarea debemos poder finalizar la tarea, borrarla, reasignarla, cambiar prioridad, fecha, hora, nombre y su descripción.



Como lo hace Asana

☒ Marcar como finalizada

Esta tarea es privada para los miembros de este proyecto.
 Hacer público

Junio 06, 2023 Revisar sistema

Responsable Andres Lara Asignadas recientemente

Fecha de entrega Hoy

Proyectos Zavir Rentas Reuniones

Agregar a proyectos

Descripción

DESCRIPCION DE LA TAREA
 Escribe "/" para ver el menú

FORMATO DE TEXTO

+ | **B** | *I* | U | | | | | | | |

Subtareas

	Lista de equipo menor en Dashboard para saber cuantos dias y la fecha que vence la renta, con colores, que incluya cliente y numero de nota de venta.		
	Reporte Resumen de ventas, el formato tal cual lo tienen ellos.		
	Verificar la validacion del limite de credito al momento de hacer la nota de venta.		

Nota: Seria buena idea cargar el formulario con la información de la tarea y que se pueda editar o crear en el mismo formulario y ahí mismo la ve el usuario responsable, nosotros podemos usar un botón guardar o si queremos simplificar la acción podemos llamar Guardar cada que se modifique un control.

- *Una tarea puede tener Subtareas. Para la sección de subtareas estaba considerando dos opciones, relistar solo las subtareas cuando la seleccionamos o mostrar una lista nueva de subtareas debajo del panel de la tarea padre.
- Progreso por tarea. Al igual que el proyecto cada tarea debe de tener un progreso, se calcula de la misma manera.

$$\text{ProgresoTarea} = \text{TareasCompletadas} / (1 + \text{TotalSubtareas}) * 100$$

Nota: El 1 representa la tarea principal, de tal manera que si la tarea principal no tiene subtareas el progreso se completara al realizar la tarea principal.