

Integrantes:
Nicolas Perez
Samuel Jimenez
Valeria Caro

Proyecto 3

Resumen:

La WBS (Work Breakdown Structure) es una estructura de las tareas que se deben llevar a cabo dentro de un proyecto y usualmente están organizadas de forma jerárquica, agrupadas por tipo de tarea, etapa o cualquier otro criterio que sea relevante. Una WBS es una herramienta de gestión que permite describir todas las tareas que se deben realizar dentro de un proyecto, planear el tiempo que tardarán, y hacerle seguimiento a su avance. A partir de ahora, su aplicación tiene que ser capaz de soportar la gestión de una WBS, teniendo en cuenta las siguientes restricciones.

Proyecto:

- Se agrega una WBS, compuesta por paquetes de trabajo y tareas organizados de forma jerárquica: un paquete de trabajo puede tener dentro otros paquetes de trabajo o puede tener tareas.
- Se agrega Tipo de Tareas

WBS:

- Compuesta por paquetes de trabajo y tareas organizados de forma jerárquica.
- Puede tener un cantidad de niveles diferentes (la raíz debe ser siempre un paquete de trabajo)

Paquete de Trabajo

- Atributos: Nombre, Descripción

Tarea:

- Atributos: Nombre, Descripción, Tipo, Tiempo Planeado, Fecha Estimada de Finalización y Responsables.
- Para cada tarea se podrá definir también una cantidad estimada de tiempo que tomará (El tiempo planeado.)
- Los tipos de tareas se definen en cada proyecto
- Una tarea podría tener varias actividades asociadas
- El tiempo real de una tarea es la suma del tiempo de las actividades que se realizaron para cumplir con la tarea.
- La fecha de finalización de una tarea es la fecha en la cual se realizó la actividad que completó la tarea.

Actividades:

- Cuando se registre una actividad realizada en un proyecto, debe indicarse la tarea dentro de la WBS a la que quedará asociada.
- Al asociar a una actividad a una tarea debe indicarse si con esa actividad se completa la tarea o no.

Restricciones:

- No se pueden eliminar tareas que ya tengan actividades relacionadas

Roles

- Proyecto
- WBS
- Paquete de Trabajo
- Tareas
- Actividades

Responsabilidades

- Agregar actividades a una tarea
- Agregar WBS a proyecto
- Agregar el tipo de tareas
- Indicar la tarea de la que la WBS quedará asociada

Colaboraciones

1. Agregar actividades a una tarea (Actividades, Tarea)
2. Agregar WBS a proyecto (WBS, Proyecto)
3. Agregar el tipo de tareas (Proyecto, Tareas)
4. Indicar la tarea de la que la WBS quedará asociada (Actividades, Tarea)

WBS:

Para implementar la WBS en nuestro proyecto hemos decidido crear tres clases:

Clase Paquete de trabajo:

Parámetros:

- Nombre
- Descripción

Clase Tareas:

Parámetros:

- Nombre
- Descripción
- Tipo
- Fecha estimada
- Responsables

Clase WBS:

- Raíz paquete de trabajo

Para la implementación del WBS decidimos optar por crear un árbol ya que pensamos que esta era la manera mas optima para implementar este sistema en nuestra aplicación puesto que con un árbol podemos crear una raíz (paquete de trabajo) que tenga nodos hijos

(paquetes de trabajo o tareas). Algo parecido al Taller 7, ya que en este también se creaba un árbol con una raíz a la cual se le podían agregar nodos de diferentes categorías.

Reportes:

Para los reportes decimos agregar otra clase en la parte de lógica que se encargará de sacar la información necesaria para graficar los reportes

Clase Reporte

- Avance del proyecto
- Calidad de la planeación (Tiempo real/ tiempo estimado), ayudar con esto a dar tareas mal planeadas
- Desempeño de equipo, individual y grupal
- Resumen general del proyecto

Interfaz Gráfica:

Para la interfaz gráfica hemos decidido agregar dos paneles y tres frames.

PReport: En PReport se realizarán los reportes necesarios y la idea es poder mostrarlos como gráficas usando librerías como XChart.

PWbs: El PWBS será la interfaz encargada de manejar los paquetes de trabajo y las tareas.

PPackage: Es un frame donde se crea la información del paquete

PTask: Es un frame donde se crea la información de la tarea

PMTask: Es un frame donde el usuario podrá seleccionar la tarea que desea modificar

Problemas para considerar:

- Como estamos implementando más clases debemos ser conscientes de que como estamos aplicando un Modelo MVC, hay que seguir la misma línea para crear las funcione y demás.
- Al implementar la función de eliminar tareas hay que ser cuidadosos respecto a qué estructura de datos utilizar, ya que puede que haya formas de no complicarnos mucho con la implementación

Excepciones:

Es importante recalcar que tener en cuenta las excepciones que pueden ocurrir en el proyecto nos permite que los usuarios no cometan errores y que precisamente la aplicación tampoco. Algunas de las excepciones que manejamos a lo largo del proyecto fueron:

- Cuando se inicializa la aplicación el usuario debe crear un usuario y seleccionar un proyecto o crear un proyecto porque si no la aplicación por medio de un JDialog le mostrará al usuario una advertencia de que no puede ir al panel debido a que se ha saltado un paso
- Cuando el usuario crea una actividad, proyecto, tarea o paquete de tarea debe rellenar toda la información que se le pide, ya que la aplicación verifica que no haya espacios vacíos, si se encuentra por lo menos con uno manda una excepción/mensaje de que falta información por ingresar
- Al crear participante el usuario debe tener en cuenta que no se puede crear dos participantes con el mismo nombre, sino esto mandará una advertencia de que ya existe un usuario con ese nombre.
- Teniendo en cuenta lo anterior, también se debe aplicar cuando se cree una tarea ya que si se crea con el mismo nombre y tipo la aplicación puede que falle, mientras que, si se crea una tarea con el mismo nombre, pero con diferente tipo se evitará esta falla. Para esto también se va a crear una advertencia que pueda ser mostrada al usuario.
- Por ejemplo, cuando el usuario desea modificar una actividad debe ser una a su nombre porque si no la modificación no se realizará, para ello se le lanza una advertencia al usuario para que tenga en cuenta esto.
- Otra de las excepciones tenida en cuenta fue a la hora de crear un proyecto, ya que antes de crearlo la aplicación envía un mensaje al usuario de que ingrese bien la fecha ya que nuestra aplicación usa un formato específico y si el usuario la ingresa mal por ende la aplicación no recogerá esa información de forma correcta.

Reflexión

- Poder tener una implementación desacoplada nos permite trabajar con facilidad entre la lógica y la vista, esto ha hecho que implementar nuevas funcionalidades a la aplicación sea menos complicado y sin tantas dependencias. También es importante mantener comentarios en el código ya que al empezar a agregar más clases probablemente vayamos olvidando lo que hicimos en una o en otra.

Por último, en el repositorio se encontrarán diagramas con lo anteriormente mencionado en el documento.