

Ejercicio 2. Arreglos básicos

Análisis

Requisitos funcionales

- Crear nuevo proyecto
- Agregar tarea a proyecto
- Crear nuevo desarrollador
- Actualizar estado de tarea
- Generar reporte de progreso
- Opción de salir del programa
- En cada proyecto se debe de asignar un desarrollador Junior y un desarrollador Senior
- A cada proyecto se le debe de definir a que categoría pertenece
- El programa solo debe de permitir códigos distintos a los ya creados de proyectos
- Se debe de asignar fecha límite a cada proyecto
- En cada tarea se le debe de definir a que tipo pertenece
- El programa debe de permitir que los desarrolladores de cada tarea sea únicamente los desarrolladores del proyecto
- A cada tarea se le debe de asignar un tiempo determinado
- Cada vez que un desarrollador trabaje, debe de actualizar el tiempo real trabajado

Propósito de cada clase

Proyecto: Esta clase tiene como propósito almacenar la información del proyecto creado.

Tarea: Esta clase tiene como propósito almacenar la información de la tarea creada.

Desarrollador: Esta clase tiene como propósito almacenar la información del desarrollador creado.

Main: Esta clase se encargará de realizar los métodos del programa.

Interfaz: Se encargará de mostrar lo necesario al usuario y que el usuario pueda ingresar los datos.

Propósito de atributos

Proyecto:

Nombre: String. Se utilizará para definir el nombre del proyecto que se está creando.

Categoría: int. Se utilizará para definir que categoría tendrá el proyecto que se esta creando, entre desarrollo y mantenimiento.

Prioridad: int. Se utilizará para definir que prioridad tendrá en proyecto que se esta creando, entre alta, media y baja.

DesarrolladorJ: desarrollador. Se utilizará para definir el desarrollador Junior que tendrá el proyecto que se está creando.

DesarrolladorS: desarrollador. Se utilizará para definir el desarrollador Senior que tendrá el proyecto que se está creando.

Fecha: Date. Se utilizará para definir la fecha limite que tendrá la entrega del proyecto que se está creando.

tareas: Tarea[]. Lista de tareas asociadas al proyecto.

Tarea:

Tipo: int. Se utilizará para definir que tipo de tarea será la tarea que se está creando, entre desarrollo, pruebas y documentación.

Desarrollador: desarrollador. Se utilizará para definir el desarrollador que tendrá la tarea que se está creando.

Testimado: double. Se utilizará para definir el tiempo estimado de la tarea que se está creando.

Treal: double. Se utilizará para que el desarrollador asignado a la tarea pueda agregar el tiempo trabajado en la tarea.

Nombre: string. Se utilizará para definir el nombre de la tarea que se está creando.

Estado: int. Se utilizará para que el desarrollador asignado a la tarea pueda definir el estado de la tarea.

Desarrollador:

Puesto: int. Se utilizará para definir el puesto del desarrollador que está creando.

Nombre: string. Se utilizará para definir el nombre del desarrollador que se está creando.

Contrasenía: String. Se utilizará para colocarle una contraseña para ingresar a ese desarrollador.

Main:

proyectos: Proyecto[]. Lista de todos los proyectos gestionados.

desarrolladores: Desarrollador[]. Lista de todos los desarrolladores disponibles.

Interfaz:

Main: Main. Se utilizará para poder realizar los métodos necesarios.

Propósito de métodos

Proyecto:

agregarTarea(tarea: Tarea): Añade una nueva tarea al proyecto.

Main:

agregarProyecto(proyecto: Proyecto): Añade un nuevo proyecto a la empresa.

agregarDesarrollador(desarrollador: Desarrollador): Añade un nuevo desarrollador a la empresa.

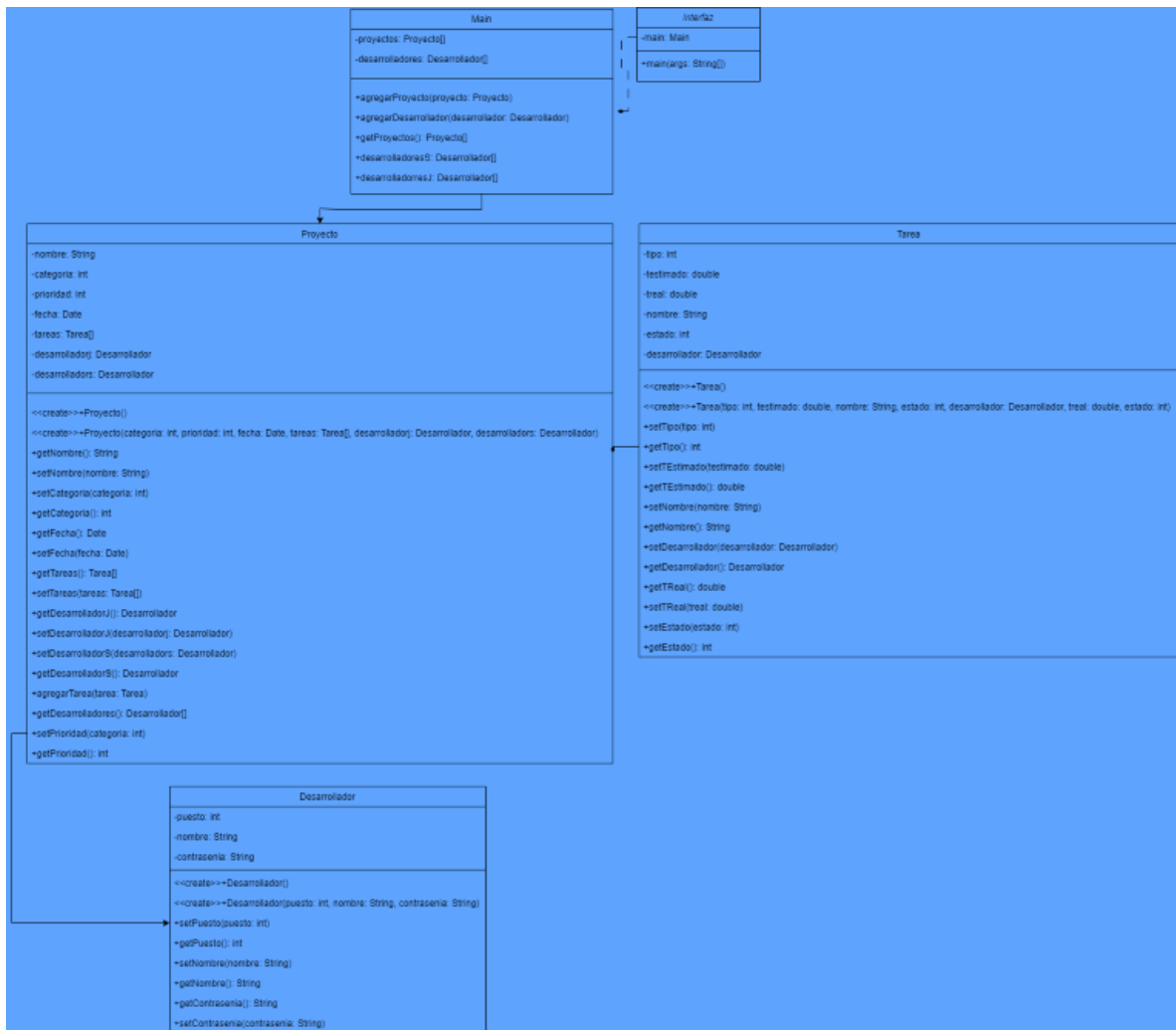
desarrolladoresS(): Devolvera los desarrolladores Senior-

desarrolladoresJ(): Devolvera los desarrolladores Junior

Interfaz:

Main(args String[]): Se utilizará para mostrar al usuario lo necesario.

Diseño



Se modificó para que se tenga un mayor orden y un mejor uso de métodos. Así como implementar seguridad con una contraseña.