

Capítulo 1

Diario de Desarrollo

Este capítulo va a ser provisional en la memoria y solo se va a utilizar para plasmar el avance del desarrollo del proyecto. Como ya tenemos una lista de las diferentes partes que vamos a desarrollar, a continuación entregamos una lista con las tareas realizadas y otra con las tareas a realizar, además de algún comentario si se ve la necesidad de esclarecer algo más.

1.1. Tareas Realizadas

- Editor de Código:
 - Desarrollo de la estructura principal.
 - Inclusión de funciones de edición habituales.
 - Desarrollar compatibilidad para usar diferentes ventanas al mismo tiempo.
 - Integración de resaltado de sintaxis, resaltado de línea y guía de filas.
 - Integración de autocompletado, analizador sintáctico.
 - Integración del módulo completo en el programa.
- Gestor de Proyectos
 - Desarrollo de la estructura principal.
 - Desarrollo de las funciones del módulo (Crear, Usar, Navegar, Guardar Proyectos y Archivos).
 - Mejorar la vista de proyectos.
 - Integración del módulo completo en el programa
- Gestor de Plantillas
 - Desarrollo de la estructura principal.
 - Desarrollo de las funciones del módulo (Crear, Borrar y Usar plantillas).
 - Integración del módulo completo en el programa.
- Compilación
 - Integración del Java Compiler API en el programa.
 - Implementación de compilación de proyectos.
- Ejecución

- Implementación ProcessBuilder y comprobar que ejecute adecuadamente.
- Desarrollo de la posibilidad de una ejecución paso a paso.
- Integración del módulo completo en el programa
- Depuración
 - Implementación de JPDA/JDI en el programa.
 - Integración de varias ayudas visuales
 - Integración del módulo completo en el programa
- Integración de Herramientas Externas
 - Desarrollar estructura principal del módulo.
 - Implementar algunos módulos o APIs como JGit.
 - Integración del módulo completo en el programa
- Configuración del Entorno
 - Desarrollar la capacidad de tener archivos y ajustes personalizados.
 - Desarrollar la admisión de nuevos aspectos para la interfaz.
 - Comprobar e implementar las herramientas disponibles en el framework.
 - Integración del módulo completo en el programa
- Interfaz
 - Creación de la estructura principal, iconos y ventanas.
 - Integración de los diferentes módulos desarrollados en la interfaz.
- Implementación IA
 - Desarrollo de la interfaz del chat.
 - Integración del modelo IA Qwen con Ollama.
 - Implementación e integración en el programa.
 - Memoria
 - Aprender a usar \LaTeX .
 - Crear una primera versión.

1.2. Tareas por realizar

- Interfaz
 - Integración de cada módulo en la interfaz.
- Compilación

- Desarrollo del modo de compilación incremental.
- Integración del módulo completo en el programa
- Ayuda
 - Búsqueda y creación de manuales apropiados.
 - Búsqueda y creación de código de ejemplo.
 - Creación de tutoriales para el uso del IDE.
 - Integración del módulo completo en el programa
- Kubernetes
 - Configuración de Minikube
 - Creación de contenedores para los módulos.
 - Despliegue de los servicios Kubernetes
- Memoria
 - Actualizar esta según vamos desarrollando el IDE.
 - Revisión de la memoria para entrega.

1.3. Repositorio Github

Repositorio GitHub en [este enlace](#) para que se pueda ver el código y el avance en tiempo real.

1.4. Guía de Uso Simple IDE

El ZIP recibido contiene el código fuente y ejecutable del IDE. Para inicializarlo, extraiga el ZIP donde prefiera. Ahora tendrá una nueva carpeta llamada ide. Abra esta carpeta, abra la siguiente carpeta llamada también ide y abra la carpeta bin. Dentro de esta última, tendrá dos archivos .exe diferentes. Elija el que se ajuste a su versión de Windows (32 o 64 bits).

Una vez iniciado, le saltará un explorador de archivos en el que tendrá que elegir una carpeta para que sea su Workspace. Esta información junto con otros datos importantes para el funcionamiento del programa. Esta información se guarda en la ruta C:/Users/SuUsuario/.Simple-IDE. Si en algún momento quiere cambiar la configuración, volver a iniciar de cero o eliminar esta carpeta, puede hacerlo en cualquier momento.

Para probar la IA puede instalarse Ollama y qwen2.5-coder de 4GB. También hay opciones de probarlo en el navegador para que vea lo que más o menos puede esperar del chat en el editor.

En la barra superior están las opciones de Git y Debug. Para crear Breakpoints para el debug, debe clicar dos veces en el editor con un archivo abierto, en el número de línea hasta que vea un icono. A partir de ahí ya puede usar las diferentes acciones de debug que incluye el IDE.

Aparte de lo ya explicado, se comporta como un IDE normal. Si no ve los iconos para crear carpeta, crear fichero o crear paquete, por favor vaya a la opción View en la zona superior del programa, a Toolbars y marque la casilla File. Ahora ya podrá probar lo que le ofrece esta versión del programa.

También avisar de que hay botones que no funcionan y asperezas que retocar.

1.5. Resumen

1.5.1. Actualización de Julio

Como se dice arriba, este texto es simplemente para información, no formará parte de la memoria final.

La memoria esta siendo actualizada. Esto es simplemente para ver como va el desarrollo del proyecto. Como podemos observar, ya queda poco de desarrollo como tal, queda la implementación en Kubernetes y la creación de la sección de ayuda, además de la compilación incremental. Una vez estás estén terminadas, tendremos el ultimo prototipo del IDE.

También le dejo la versión del IDE tal y como está en el momento, por si quiere ojear alguna funcionalidad o apariencia. Las instrucciones están arriba, en la guía de uso. Para cualquier duda, puede contactar conmigo por correo.