

EZJAVACODE

PROPUESTA DE PROYECTO

RUBÉN MATAMOROS TRIGO

Índice.

1. Título y descripción general del proyecto.	3
2. Identificación del proyecto.....	4
3. Justificación y Objetivos.	5
4. Contenidos y aspectos principales.	7
5. Medios que se utilizarán.....	8
6. Áreas del ciclo formativo.....	9

1. Título y descripción general del proyecto.

EZJavaCode es un programa orientado a facilitar y amenizar tareas diarias para los programadores en Java haciendo parcialmente o en su totalidad bloques de código para las clases del software a desarrollar.

La aplicación utiliza bloques preestablecidos donde el usuario con simples clicks podrá crear Clases, atributos, valores, funciones, métodos, implementación de librerías...

La idea principal es que el usuario con un simple botón puede crear una clase en Java, posteriormente introducirá el nombre y será creada automáticamente.
Ejemplo:

Botón [Nueva Clase](#) -> [Introduce el nombre de la clase](#).

Para este ejemplo el usuario introducirá en el campo de nombre de la clase el nombre ["Persona"](#)

Una vez con la clase creada podrán incluir atributos dentro de la clase con un menú selector de "Tipo" de atributo (String, Integer, Double, Boolean...), a continuación el usuario definirá el nombre del atributo y si es necesario el valor del mismo como se muestra en el siguiente ejemplo.

Tipo / Nombre Atributo / Valor del campo.

[String](#) / [Nombre](#) / [Rubén](#)

Este paso podrá ser repetido tantas veces como el usuario necesite, cabe recalcar que no se incluyen solo "Tipos" predefinidos, también cuando el usuario cree sus nuevas clases en el desplegable selector de "Tipo" aparecen también sus clases, como idea a futuro que el usuario entre varios programas pueda almacenar todas las clases que él desee.

Una vez el usuario haya terminado con su clase se le dará la posibilidad de exportar esa clase, a un fichero .java.

En cuanto al tema de funcionalidades, la idea es con las funcionalidades en bloque añadir indefinidamente diferentes códigos preestablecidos.

Por ejemplo Condicionales, funciones auto agregadas para recorrer listas, escritura de archivos...

La idea es que mediante Actualizaciones el proyecto sea infinitamente escalable. Esto abre un abanico de posibilidades cara a desarrollo a futuro.
Ejemplo:

Al usuario le interesa implementar la librería "Math", gracias al sistema de bloques sería relativamente sencillo la implementación de la misma o de cualquier otra librería o bloques de código preestablecidos.

2. Identificación del proyecto.

Autor del proyecto:
Rubén Matamoros Trigo

Ciclo formativo:
Grado Superior en Desarrollo de Aplicaciones Multiplataforma.

Centro educativo:
IES Albarregas. Mérida.

3. Justificación y Objetivos.

3.1. Justificación.

Desarrollar software en Java, especialmente en las primeras etapas, puede ser un proceso tedioso si no se cuenta con una buena organización del código. Definir clases, atributos y métodos manualmente no solo consume tiempo, sino que también aumenta las posibilidades de cometer errores sintácticos o inconsistencias en la nomenclatura.

Durante mi formación en el Grado Superior de Desarrollo de Aplicaciones Multiplataforma (DAM), me di cuenta de que, aunque existen herramientas como los paquetes (packages), librerías externas (Math, JUnit, Gson...) o los Live Templates, muchas tareas siguen dependiendo de una escritura manual repetitiva. Esto no sólo ralentiza el desarrollo, sino que también afecta la productividad.

De esta necesidad nació la idea de EZJavaCode, una aplicación pensada para agilizar y optimizar la escritura de código en Java mediante bloques preestablecidos. Con esta herramienta, los programadores pueden generar clases, definir atributos, establecer métodos y añadir librerías de forma más intuitiva, reduciendo errores y acelerando el proceso de desarrollo.

EZJavaCode no solo es útil para desarrolladores con experiencia, sino que también facilita el aprendizaje de quienes están empezando, gracias a su interfaz visual e interactiva. Además, al permitir la exportación de clases en archivos .java, se integra sin problemas con cualquier entorno de desarrollo.

Otro punto clave de la herramienta es su escalabilidad. Está diseñada para crecer con nuevas funcionalidades, como la integración de más librerías, la automatización de ciertas tareas y la generación de estructuras de datos más avanzadas. Esto garantiza que EZJavaCode pueda evolucionar y adaptarse a las necesidades cambiantes de los desarrolladores.

En resumen, EZJavaCode busca hacer que la programación en Java sea más eficiente, accesible y organizada, tanto para quienes ya tienen experiencia como para quienes están aprendiendo.

3.2. Objetivos.

Objetivo general

Desarrollar una aplicación multiplataforma que facilite la generación de código en Java mediante una interfaz interactiva basada en bloques predefinidos, permitiendo la creación rápida y estructurada de clases, atributos, métodos, librerías... con la posibilidad de exportar el código generado en archivos .java.

Objetivos específicos

3.2.1. Optimización del flujo de trabajo en Java.

- Reducir el tiempo de desarrollo al minimizar la escritura manual de código mediante un sistema de generación automática de clases y métodos.
- Implementar patrones de diseño que garanticen la coherencia estructural del código generado.

3.2.2. Reducción de errores humanos.

- Disminuir errores sintácticos y de lógica mediante la generación de código estructurado basado en plantillas predefinidas.
- Establecer validaciones para evitar la declaración incorrecta de atributos y métodos.

3.2.3. Interfaz visual e intuitiva.

- Desarrollar una interfaz gráfica que permita la creación de código de manera visual, sin necesidad de conocimientos avanzados en Java.
- Implementar un sistema de drag & drop para facilitar la selección y edición de bloques de código.

3.2.4. Facilitación del aprendizaje en Java.

- Proporcionar una herramienta didáctica para estudiantes y principiantes en programación orientada a objetos.
- Permitir la visualización en tiempo real del código generado a partir de los bloques seleccionados.

3.2.5. Escalabilidad y modularidad.

- Diseñar la arquitectura de la aplicación de forma modular para permitir la integración de futuras funcionalidades, como nuevos bloques de código.
- Incorporar la posibilidad de almacenamiento y reutilización de clases y métodos creados por el usuario en diferentes proyectos.

3.2.6. Compatibilidad con entornos de desarrollo.

- Permitir la exportación de código en archivos .java para su uso en IDEs como IntelliJ IDEA, Eclipse y NetBeans.
- Asegurar que el código generado siga las convenciones y buenas prácticas de Java, facilitando su integración en proyectos reales.

4. Contenidos y aspectos principales.

EZJavaCode se centra en la automatización de la generación de código en Java mediante bloques interactivos, proporcionando a los usuarios las siguientes funcionalidades principales:

Creación rápida de clases: Generación automática de clases en Java con solo introducir un nombre.

Definición de atributos y métodos: Selección de tipos de datos predefinidos (String, int, double, boolean, etc.) y clases personalizadas creadas por el usuario.

Implementación de librerías: Inclusión sencilla de paquetes estándar como Math, Scanner, ArrayList, entre otros, con opciones para futuras ampliaciones.

Estructuras de control predefinidas: Bloques para condicionales (if-else), bucles (for, while), y manejo de excepciones (try-catch).

Visualización en tiempo real: Conversión instantánea de los bloques en código Java listo para exportar.

Exportación de código: Posibilidad de guardar las clases creadas en archivos .java para su integración en cualquier IDE.

Escalabilidad y almacenamiento: Potencial para que los usuarios almacenen y reutilicen sus propias clases en proyectos futuros.

Estas funcionalidades están diseñadas para mejorar la productividad de los programadores, reducir errores y hacer que la escritura de código sea más intuitiva.

5. Medios que se utilizarán.

Para el desarrollo de EZJavaCode, se utilizarán las siguientes herramientas y tecnologías:

Entorno de desarrollo: El proyecto se desarrollará en IntelliJ IDEA, un IDE ampliamente utilizado para proyectos en Java.

Lenguaje de programación: Se utilizará Java, asegurando compatibilidad con cualquier entorno de desarrollo.

Interfaz gráfica: Se contempla el uso de JavaFX para la creación de una interfaz interactiva y amigable.

Almacenamiento de datos: Dependiendo de la evolución del proyecto, podría evaluarse el uso de bases de datos locales como SQLite para la gestión de clases personalizadas creadas por el usuario.

Exportación y compatibilidad: Se asegurará que el código generado sea compatible con IDEs como IntelliJ IDEA. Una opción a evaluar es la creación de un proyecto vacío donde los usuarios podrán tener ya organizado su propio proyecto.

EZJavaCode se plantea como una solución escalable, adaptable a futuras mejoras y con potencial para expandirse con nuevas funcionalidades en el futuro.

6. Áreas del ciclo formativo.

Asignaturas / Conocimientos del primer año:

Programación.

Entornos de desarrollo.

Bases de datos.

Asignaturas / Conocimientos del segundo año:

Acceso a datos.

Desarrollo de interfaces.

Programación de servicios y proyectos.