

Primera parte

Realizar una investigación sobre cuáles son los IDES existentes en el proceso de creación código, sus diferentes versiones y las ventajas y desventajas de cada uno de ellos.

Eclipse

- Ventajas
 - Compilador customizable
 - Integración fácil con Git
 - Mercado de plugins
 - Auto-completación de código
 - Refactorizar
- Desventajas
 - Fama de ser lento
 - Extenso uso de CPU
 - Herramienta pesada con frecuentes crashes
 - Complicado entorno de plugins
- Versiones
 - Che: Eclipse Che is a developer workspace server and cloud IDE
 - Theia: An open, flexible and extensible cloud & desktop IDE platform
 - Jetty: Eclipse Jetty provides a web server and javax.servlet container
 - Eclipse GlassFish: Eclipse Jetty provides a web server and javax.servlet container Enterprise

NetBeans

- Ventajas
 - Entorno simple de plugins
 - Comparación de código lado-a-lado
 - Análisis de código fuente para la ayuda de mejorar el código
 - Uso móvil
 - Soporte Maven, GIT
- Desventajas
 - Lento debugger
 - Alto consumo de memoria
- Versiones
 - Apache NetBeans 19.0 (Septiembre 2023), versión única
 - Mejoras: <https://github.com/apache/netbeans/releases/tag/19>

IntelliJ IDEA

- Ventajas
 - Versión de comunidad (open-source), versión propietaria (pago)
 - Fuerte modificación de elementos
 - Soporte para otros programas basados en JVM como Kotlin
 - Soporta varios lenguajes
 - Soporte de control de versión integrada
 - Fuerte integración y soporte de plugins
 - Sistema de Gradle
- Desventajas
 - Alto precio de versiones mas altas
 - Alto consumo de recursos de sistema

- Versiones:
 - Fleet: Ligero y rápido
 - IntelliJ IDEA: Principal IDE
 - Aqua: Para automatizar tests
 - DataGrip: Para bases de datos (pago)
 - DataSpell: Análisis de datos

BlueJ

- Ventajas:
 - Simple GUI
 - Habilidad para invocar expresiones Java
 - Interactuar entre objetos
 - Muestra gráficos UML en proyectos
- Desventajas
 - Frecuentes crashes
 - Bueno para principiantes pero escasea en elementos para aplicaciones robustas
 - 32-bits
- Versiones
 - 5.2.1 (Octubre 2023) Versión única

Jdeveloper

- Ventajas
 - Fuerte integración con “desarrollo ágil” y programas de versiones
 - Completo mantenimiento de la vida del programa SDLC
 - Fuerte conexiones con bases de datos
 - Habilidad para ejecutar comandos SQL
- Desventajas
 - Demasiada alta la curva de aprendizaje inicial
 - Alto consumo de RAM
 - Pesado y puede ralentizar la ejecución de programas
- Versiones
 - Sutidio Edition 12.2.4.0: Versión completa con todos los elementos
 - Java Edition 12.2.1.4.0: Versión delgada y mas ágil

Segunda parte

Los IDEs encontrados en la primera parte de la actividad son utilizados para realizar programas que tienen diferentes funcionalidades, pero antes de ponernos a programar, deberíamos realizar un estudio de cuáles son las necesidades que tiene el programa que vamos a desarrollar, cuáles son los datos que necesitamos, cual es el flujo de ejecución del programa, etc.... Para esta parte de la actividad se pide que describáis los siguientes conceptos:

- Algoritmo
- Pseudocódigo
- Diagramas de flujo
- Datos entrada / salida

- Algoritmo: conjunto ordenado y finito de operaciones simples a través del cual podemos hallar la solución a un problema
- Pseudocódigo: forma de expresar los distintos pasos que va a realizar un programa, de la

forma mas parecida a un lenguaje de programación. Su principal función es la de representar por pasos la solución a un problema o algoritmo, de la forma mas detallada posible, utilizando un lenguaje cercano al de la programación.

- Diagramas de flujo: diagrama que describe un proceso, sistema o algoritmo informático. Tienen amplios y numerosos campos de aplicación pero principalmente se usan para comunicar procesos que suelen ser complejos y convertirlos diagramas claros y fáciles de entender.
- Datos entrada / salida: conjunto de acciones necesarias para la transferencia de un conjunto de datos, es decir, una transferencia completa de todos los datos. . Así, el subsistema de entrada / salida suministra al ordenador un mecanismo eficiente de comunicación entre el procesador central y el entorno exterior.

Tercera parte

Una vez tenemos claros todos los conceptos anteriores, ahora vamos a realizar el algoritmo de un proceso cotidiano como por ejemplo hacer la cena, jugar una partida de ajedrez, lavar el coche, etc... Para la entrega deberéis de hacer tanto el pseudocódigo como el diagrama de flujo correspondiente a la tarea

El documento a entregar tendrá formato pdf con el siguiente nombre:

nombre_apellido.pdf (poner vuestro nombre y vuestro apellido)

Pseudocódigo para hacer la cena:

```
cena terminada = falso
no hay ingredientes necesarios
si no hay ingredientes:
    comprar ingredientes
    traer ingredientes a casa
de lo contrario: // Comenzar a cocinar
    trocear ingredientes
    agua hirviendo = falso
    hevir agua
    mientras el agua no hierve:
        esperar
    echar todos los ingredientes en crudo
    esperar 1 hora
    si cena terminada == falso:
        esperar
    cena terminada = verdadero
```

