

Índice

Contenido

Índice	1
Requisitos de la tarea:	1
Equipos / IDEs. Team leader	2
Visual Studio CODE	2
Netbeans	2
Eclipse	3
Android Estudio	3
Intelli J	3
Visual Studio Community	3
Ejemplo de planteamiento / Organización de trabajo	3
Actividad IDE 6 sesiones de trabajo en equipo usando Trello y Google	3
Sesión 1: Introducción y asignación de tareas	4
Sesión 2: Investigación inicial y preparación del contenido	4
Sesión 3: Ejemplos prácticos y funcionalidades del IDE	4
Sesión 4: Extensiones y control de versiones	5
Sesión 5: Revisión final y simulacro de presentación	5
Sesión 6: Presentación final	5
Ejemplo de estructura en Google Slides	6
Herramientas utilizadas	6
Fecha de entrega: miércoles 4 diciembre 14horas	6
Algunas características de los IDEs:	7

Requisitos de la tarea:

IMPORTANTE 1º Desinstalar el IntelliJ Ultimate.

Con esta actividad aprenderemos sobre algunos de los **Entornos de Desarrollo Integrados** y se valorará el **trabajo en equipo** ([*Qué son las soft skills para el trabajo en equipo*](#)).

Recordad la metodología ágil: el tiempo apremia.

Describe el entorno de desarrollo que os ha tocado explicando cada uno de los siguientes aspectos:

ACTIVIDAD EN EQUIPO UD2: ENTORNOS DE DESARROLLO INTEGRADOS (IDEs)

- Breve introducción definiendo el Entorno (origen, historia, principales características)
- Explicación sobre descarga e instalación del entorno (el entorno ya deberá estar descargado en el pc de presentación)
- Explicación de las distintas secciones/menús/perspectivas
- Explicar y ejemplificar las posibilidades de customización (apariencias, asistente y generación de código, etc)
- Ejecución de un programa de ejemplo en Java (el programa deberá contener funciones y comentarios)
- Función de autocompletado
- ¿Permite introducir automáticamente tabulaciones y espaciados para aumentar la legibilidad?
- ¿Detecta errores de sintaxis en tiempo real?
- Comprobar y ejemplificar si ofrece refactorización de código: cambios menores en el código que facilitan su legibilidad sin alterar su funcionalidad (por ejemplo cambiar el nombre a una variable).
- Establecimiento de puntos de ruptura con algún ejemplo
- Generación de documentación
- Instalación y desinstalación de complementos. Explicación de alguno de ellos por encima.
- Control de versiones: comprobar si se mantiene un historial local de archivos que han cambiado con versiones anteriores y ejemplificarlo
- Sistema de actualización del entorno

¿Cómo se realiza la actividad?

1. Preparación presentación con Google Drive con la explicación de cada uno de los puntos (de la manera más descriptiva posible/evitar mucho texto).
2. Presentación en clase en 20 min de la presentación, ejemplificando cada uno de los puntos en el propio entorno.

Nota: las presentaciones deberán adjuntarse en enlace y en PDF antes de la fecha límite. Definiremos días de presentación.

Nota 2: el trabajo es en grupo y debe quedar claro tanto en la presentación en clase como en el trabajo previo que todas las personas del grupo han aportado para realizar la actividad.

Equipos / IDEs. Team leader

Visual Studio CODE

1. Abad*****, Dylan
2. Arév*****, Francisco José

3. Here*****, Juan Carlos
4. Luna*****, Laura
5. Cast*****, José Antonio

Netbeans

ACTIVIDAD EN EQUIPO UD2: ENTORNOS DE DESARROLLO INTEGRADOS (IDEs)

1. Adam*****, Javier
2. Ariz*****, José Antonio
3. **Delg*****, Antonio**
4. Lagu*****, Manuel
- 5.

Eclipse

1. **Agui*****, Sergio**
2. Blan*****, Manuel
3. Fern*****, Isaac
4. Lópe*****, Francisco
5. Mont*****, David

Android Estudio

1. **Cast*****, Pedro**
2. Gali*****, Jesús
3. Guer*****, Javier

4. More*****, Alejandro Jesús
5. Nada*****, Pedro Antonio

Intelli J

1. Cast*****, Daniel
2. Garc*****, Francisco Javier
3. Pant*****, Juan Rafael
4. **Quev*****, Miguel Ángel**
5. Roja*****, Carolina

Visual Studio Community

1. Sole*****, Francisco Javier
2. Serr*****, Jesús
3. Sosa*****, Kleopatra Haydee
4. Tori*****, Miguel
5. **Vale*****, Rafael**

Team Leader: Coordina, organiza, crea tablero Trello, crea Presentación Google y me pregunta las necesidades y problemáticas de cada equipo.

Ejemplo de planteamiento / Organización de trabajo

Actividad IDE 6 sesiones de trabajo en equipo usando Trello y Google

La actividad se distribuye en 6 sesiones de trabajo colaborativo para investigar y presentar un IDE (Entorno de Desarrollo Integrado. Como equipo, no tenéis que adherirse estrictamente al siguiente planteamiento de ejemplo.

Sesión 1: Introducción y asignación de tareas

- **Objetivo:** Familiarizarse con la actividad, los puntos a abordar y la plataforma Trello.
 - **Tareas:**
 1. Crear un tablero en Trello para cada equipo con las siguientes columnas:
 - **Por hacer:** Tareas iniciales.
 - **En progreso:** Tareas que están siendo trabajadas.
 - **Revisión:** Tareas finalizadas y pendientes de revisión.
 - **Hecho:** Tareas completadas.
 2. Dividir los puntos a investigar entre los integrantes del equipo en Trello.
 - Ejemplo de tarjetas:
 - "Breve introducción al IDE"
 - "Descarga e instalación"
 - "Secciones/menús/perspectivas"
 - "Customización del entorno"
 - ...
 3. Crear una presentación en Google Slides colaborativa.
 4. Asignar responsables para la descarga del IDE y la instalación en el equipo.
-

Sesión 2: Investigación inicial y preparación del contenido

- **Objetivo:** Investigar los puntos asignados y completar información clave.
 - **Tareas:**
 1. Cada miembro del equipo investiga su tema asignado y añade su información a Google Slides.
 - Usar gráficos, capturas de pantalla, o ejemplos visuales si es necesario.
 2. Documentar las fuentes utilizadas y añadirlas al final de la presentación.
 3. Subir avances al tablero de Trello y marcar tareas como "En progreso" o "Revisión".
 4. Resolver dudas entre los equipos mediante un breve foro colaborativo (puede hacerse mediante Trello o un chat grupal).
-

Sesión 3: Ejemplos prácticos y funcionalidades del IDE

- **Objetivo:** Implementar y demostrar las funcionalidades del IDE en un ejemplo práctico.
- **Tareas:**
 1. **Ejecución de un programa Java:**
 - Escribir un programa en Java que incluya funciones y comentarios.
 - Probar el autocompletado, las tabulaciones automáticas y la detección de errores en tiempo real.
 2. **Ejemplificar funcionalidades:**

- Refactorización de código.
 - Personalización del entorno.
 - Generación de documentación.
 - Establecimiento de puntos de ruptura para depurar el programa.
3. Documentar con capturas de pantalla o grabaciones y añadir a la presentación.
-

Sesión 4: Extensiones y control de versiones

- **Objetivo:** Explorar complementos y el sistema de control de versiones.
 - **Tareas:**
 1. Instalar y desinstalar complementos relevantes al desarrollo en Java.
 - Explicar brevemente cómo afectan la funcionalidad del IDE.
 2. Probar si el IDE tiene soporte para control de versiones:
 - Mostrar cómo se visualizan los cambios en el historial local.
 - Explicar cómo actualizar el entorno o configurarlo para trabajar con Git.
 3. Añadir ejemplos visuales y explicaciones claras a la presentación.
-

Sesión 5: Revisión final y simulacro de presentación

- **Objetivo:** Revisar, corregir y practicar la exposición.
 - **Tareas:**
 1. Revisar el contenido de la presentación:
 - Comprobar que se abordan todos los puntos.
 - Asegurar que las explicaciones son claras y completas.
 2. Practicar la exposición en equipo:
 - Cada miembro debe explicar su parte asignada.
 - Recoger sugerencias del resto del equipo.
 3. Realizar ajustes finales en Trello y marcar todas las tareas como "Hecho".
-

Sesión 6: Presentación final

- **Objetivo:** Presentar el trabajo realizado y recibir retroalimentación.
 - **Tareas:**
 1. Realizar la presentación usando Google Slides.
 2. Cada miembro del equipo presenta su parte.
 3. Mostrar ejemplos prácticos en tiempo real (si es posible).
 4. Recoger comentarios del docente y compañeros para mejorar futuras actividades.
-

Ejemplo de estructura en Google Slides

1. **Portada:** Título del IDE y nombres de los integrantes.
 2. **Introducción:** Historia y características principales.
 3. **Descarga e instalación:** Capturas y pasos clave.
 4. **Secciones y menús:** Explicación con imágenes.
 5. **Ejemplo práctico en Java:** Código con explicaciones.
 6. **Funcionalidades destacadas:** Autocompletado, refactorización, etc.
 7. **Complementos y control de versiones:** Ejemplos de instalación y uso.
 8. **Conclusión:** Resumen de puntos clave y experiencia con el IDE.
-

Herramientas utilizadas

- **Trello:** Organización de tareas.
- **Google Slides:** Preparación de la presentación.
- **El IDE asignado:** Uso práctico y demostraciones.
- **Grabador de pantalla (opcional):** Para capturar ejemplos en el IDE.

Fecha de entrega: miércoles 4 diciembre 14horas

Características de los IDEs:

Recuerda que en la Web y en la Moodle podrás ver toda la documentación de la unidad.

Un breve resumen del propósito de los IDEs más importantes podría ser:

1. IDEs Generales (Multi-lenguaje)

Estos soportan múltiples lenguajes y son altamente configurables:

- **Visual Studio Code**
 - Popular por su ligereza y extensibilidad.
 - Compatible con cientos de lenguajes gracias a extensiones.
 - Ideal para web, backend, y scripts ligeros.
 - Gratuito.
- **Eclipse**
 - Usado especialmente para Java, pero soporta otros lenguajes como C++ y PHP.
 - Extensible mediante plugins.
 - Gratuito y de código abierto.
- **IntelliJ IDEA**
 - Reconocido por su soporte avanzado para Java, pero también incluye funcionalidades para Kotlin, Scala, y Python.
 - Enfocado en productividad y herramientas automáticas.
 - Versión gratuita (Community) y de pago (Ultimate).
- **NetBeans**
 - Ideal para Java y PHP.
 - Ligero y fácil de usar.
 - Gratuito y de código abierto.

2. IDEs para Desarrollo Web

- **PHPStorm**
 - Diseñado para PHP y lenguajes asociados como HTML, CSS y JavaScript.
 - Excelente para proyectos complejos.
 - De pago (con prueba gratuita).
- **WebStorm**
 - Especializado en JavaScript y frameworks modernos como React, Angular y Vue.js.
 - Herramientas integradas para Node.js.
 - De pago.

3. IDEs para Desarrollo Móvil

- **Android Studio**
 - El IDE oficial para desarrollo Android (Java/Kotlin).
 - Basado en IntelliJ IDEA.
 - Gratuito.
- **Xcode**

- Exclusivo para macOS, usado para desarrollar aplicaciones iOS y macOS.
- Compatible con Swift y Objective-C.
- Gratuito.

4. IDEs para Ciencia de Datos e IA

- **PyCharm**
 - Enfocado en Python, ideal para proyectos de ciencia de datos y machine learning.
 - Integración con Jupyter Notebooks y librerías como NumPy y Pandas.
 - Versión gratuita (Community) y de pago (Professional).
- **Jupyter Notebook**
 - Usado principalmente para proyectos interactivos en Python.
 - Popular en investigación y educación.
 - Gratuito.

5. IDEs para C/C++ y Desarrollo de Software de Sistema

- **CLion**
 - Diseñado para C y C++, con herramientas avanzadas para depuración.
 - De pago.
- **Code::Blocks**
 - Gratuito y ligero, orientado a C y C++.
 - Ideal para principiantes.
- **Dev-C++**
 - Otra opción gratuita y básica para C++.
 - Muy usado en educación.

6. Otros IDEs Especializados

- **Rider (C# y .NET)**
 - De JetBrains, para desarrollo en C# y .NET.
 - De pago.
- **RubyMine (Ruby y Rails)**
 - Especializado en el ecosistema Ruby.
 - De pago.
- **MATLAB**
 - Para desarrollo matemático y simulaciones.
 - Enfocado en ingeniería y academia.

Factores que Influyen en la Popularidad de un IDE:

1. **Soporte de lenguaje.**
2. **Extensibilidad y plugins.**
3. **Rendimiento.**
4. **Facilidad de uso y aprendizaje.**

ACTIVIDAD EN EQUIPO UD2: ENTORNOS DE DESARROLLO INTEGRADOS (IDES)

5. Costo (gratis vs. pago).