

LISTADO TRABAJO COLECCIONES JAVA
UNIDAD 5

	CLASE O INTERFAZ	ALUMNO 1	ALUMNO 2
1	<i>Stream 1 (Operaciones intermedias)</i>	Javi Prieto	
2	<i>Stream 2 (Operaciones terminales)</i>	Roberto Rebolledo	
3	<i>Expresiones Lambda 1 (explicación general y predicados)</i>	Sebastián Millán	
4	<i>Expresiones Lambda 2 (Interfaces funcionales)</i>	Alexander Luque	
5	<i>Trabajo con fechas /Horas LocalDate/LocalTime</i>	Alejandro Jiménez	
6	<i>Optional</i>	Mario Macho	
7	Métodos factory en Java y generación de aleatorios	Abdel Hachim	Pepe Segura
8	Queue (Colas) en java y Deque	Tomás Del Pino	
9	Inferencia de tipos en Java/var (Java 10)	Alejandro Rubens	Carlos Ruiz
10	Trabajo con cadenas (clase String, StringBuffer, StringJoiner, StringBuilder)	Manuel Molina	Pedro Franch
11	Interfaces como contrato: con métodos default y privados	Fran Ruiz	Marisa Lozano
12	Expresiones switch (Nuevas versiones de Java)	Antonio Martínez	Álvaro Polonio
13	SortedMap, TreeMap	Carlos Rasero	M. Ángel Pérez
14	Novedades Java 13, 14 y 15	Miguel Garfia	Pablo Cámara
15	Clase Collections	Rafael Hernández	Fernando Claro
16	ListIterator	José Carlos Fernández	Cristian Pulido
17	LinkedList	Daniel Martínez	Pilar Aguilar
18	Enum	Marco Pertegal	

1. TRABAJO A REALIZAR

Se trata de realizar una investigación en profundidad sobre el tema asignado, es decir, buscar materiales, consultar internet y el API de Java, leer de diferentes fuentes, ventajas y desventajas con respecto a anteriores versiones de Java (si existen en otras versiones), etc.

El trabajo debe constar de lo siguiente:

1. Un documento de texto o un videotutorial, en el que se explique para qué se usa la o las clases e interfaces y los principales métodos de la misma si tiene muchos (al menos 6 métodos si hay más de 6). No es necesario crear presentación para su exposición.

* Para la consulta o comentarios sobre el API, se debe usar la versión Java 10 o superiores, aunque la funcionalidad haya aparecido en versiones anteriores (8 ó 9).

2. Un ejemplo hecho en Eclipse bien empaquetado donde se usen adecuadamente los métodos y funcionalidades principales de la clase que se ha trabajado, incluyendo comentarios en los métodos.

MUY IMPORTANTE: El ejemplo debe estar contextualizado, es decir, debe ser como los ejercicios de clase (no es válido crear una sola clase y llamar a los métodos imprimiendo el resultado), sino que debe tener clases modelo, CRUD, principal con menú, etc.

Para ello, en la clase principal, al inicio, se debe incluir el "enunciado del ejercicio, problema o caso".

* La utilización de la clase debe quedar totalmente clara, incluyendo posibles dificultades, versiones anteriores si ahora hay algo nuevo, etc.

* Se explicarán algunas a los compañeros (Fechas, Stream y Lambda).

2. CRITERIOS DE CORRECCIÓN

Se puntuará sobre 5 puntos cada parte del proyecto, 5 para el tutorial y 5 para el ejemplo.

Originalidad/Contextualización: 1 punto.

Adecuación del contenido/profundidad en la explicación: 2 puntos.

Claridad en la explicación y ejemplos: 2 puntos.