**Universidad ORT Uruguay**

**Facultad de Ingeniería**

Aspectos de Seguridad - Obligatorio

Entregado como requisito para la obtención del crédito Aspectos de Seguridad

Segundo Semestre 2019

Marcela Ferraz - 200112

Mauricio Zito - 111725

Docentes: Santiago Paz, Felipe Sotuyo

Tabla de contenido

[Introducción 3](#_Toc26397899)

[Tecnología utilizada 3](#_Toc26397900)

[Parte 1 – Sistema seguro Bell Lapadula 4](#_Toc26397901)

[1.1 Desarrollo 4](#_Toc26397902)

[1.2 Instalación 6](#_Toc26397903)

[1.3 Prueba 7](#_Toc26397904)

[1.4 Conclusión 7](#_Toc26397905)

[Parte 2 – Canal Encubierto 8](#_Toc26397906)

[2.1 Desarrollo 8](#_Toc26397907)

[2.2 Instalación 8](#_Toc26397908)

[2.3 Prueba 8](#_Toc26397909)

[2.4 Conclusión 8](#_Toc26397910)

[Conclusión Final 8](#_Toc26397911)

# Introducción

En el marco de la materia Aspectos de seguridad, se solicita un trabajo práctico en caracter de obligatorio donde se implementará un modelo de seguridad basado en el planteado por la dupla de Billy Elliott Bell y Len LaPadula, consiste en dividir el permiso de [acceso](https://es.wikipedia.org/wiki/Acceso) de los usuarios a la información en función de etiquetas de seguridad. Por ejemplo, en sistemas militares norteamericanos, categorizándola en 4 niveles: no clasificado, confidencial, secreto y ultrasecreto.

El modelo define 2 reglas de control de acceso mandatorio (MAC):

Propiedad de seguridad simple: Un sujeto de un determinado nivel de seguridad no puede leer un objeto perteneciente a un nivel de seguridad más alto.

Propiedad \*: Un sujeto de un determinado nivel de seguridad no puede escribir un objeto perteneciente a un nivel de seguridad más bajo. (También llamada propiedad de confinamiento).

## Tecnología utilizada

El obligatorio fue implementado utilizando:

* Lenguaje Java
* IDE Netbeans versión 8.0.2

Version de Java

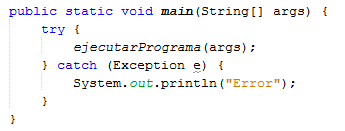
* JDK 1.8

## Parte 1 – Sistema seguro Bell Lapadula

### Desarrollo

|  |  |  |  |
| --- | --- | --- | --- |
|  | archivos | Archivos | Clase que controla la lectura y escritura de archivos (lectura de instrucciones y escritura de log). |
| Entities | InstructionObject | Clase que resume información para la ejecución del comando |
| OperationType | Enumeración de tipos de operaciones.  READ, WRITE, BAD\_INSTRUCTION |
| ReferenceMonitor | Control de instrucciones, control de relación de permiso y ejecución. |
| SecurityLevel | Enumeración de tipos de niveles de seguridad  LOW, HIGH, NOTHING |
| SecurityObject | Clase que representa a un objeto |
| SecuritySubject | Clase que representa un objeto |
| Comando | Clase que se encarga de separar parametros y verificar la sintaxis de los mismos. |
| Main | Main | Programa principal, funcion Main. |
| Management | ObjectManager | Controlador de objetos y sujetos. |

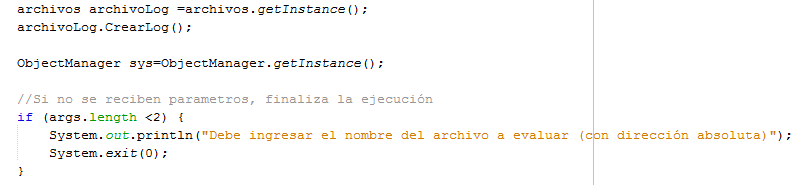
La función principal



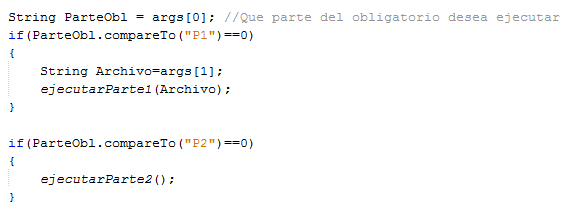
La función “ejecutarPrograma” recibe los parámetros recibidos por linea de comandos directo del main.



Varias de las funciones fueron implementadas del estilo “Singleton” para poder utilizarla de forma transversal desde diferentes secciones del proyecto.



Dependiendo del primer parámetro, se decide si se ejecuta la parte 1 o la 2 del obligatorio.



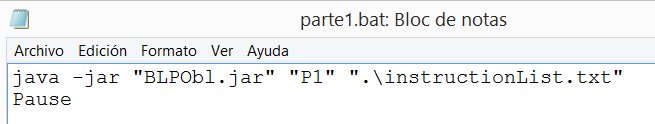


### Instalación

Para realizar la prueba, solo hay que copiar la carpeta “test” que se encuentra dentro del proyecto a cualquier carpeta del equipo destino (preferentemente Windows) .

|  |  |
| --- | --- |
|  | En la misma carpeta debe estar el archivo “InstructionList.txt” que es donde se encuentran las instrucciones a ejecutar. |

Para correr el programa se debe ejecutar utilizando el archivo “Parte1.bat” que tiene el siguiente contenido.



Parámetro “P1” indica que se va a ejecutar la parte 1 del obligatorio

### 1.3 Prueba

#### Verificación

|  |  |
| --- | --- |
| Archivo | Resultado de cada linea esperada |
|  |  |

|  |  |
| --- | --- |
| Resultado esperado | Resultado tras corrida |
|  |  |

### 1.4 Conclusión

La propuesta de la implementación de un sistema seguro con el concepto básico planteado por Bell y Lapadula fomenta la comprensión del método y las decisiones que se deben tomar para que la información vaya en un sentido seguro.

Si bien en ésta ocasión se trabajaba con números enteros, es posible implementar con documentos completos e incluso encriptados.

## Parte 2 – Canal Encubierto

### 2.1 Desarrollo

\*\* Detalle del desarrollo con fotos del código y explicación

### 2.2 Instalación

\*\* Como instalar la app en una carpeta y probarla

### 2.3 Prueba

\*\* como hacer una prueba y mostrar fotos de una prueba realizada

### 2.4 Conclusión

\*\* Conclusión de la parte 2

## Conclusión Final

\*\* Conclusión del trabajo obligatorio completo