**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](#_Toc26300204)

[Parte 1 – Sistema seguro Bell Lapadula 3](#_Toc26300205)

[1.1 Desarrollo 3](#_Toc26300206)

[1.2 Instalación 3](#_Toc26300207)

[1.3 Prueba 3](#_Toc26300208)

[1.4 Reflexión 3](#_Toc26300209)

[Parte 2 – Canal Encubierto 3](#_Toc26300210)

[2.1 Desarrollo 3](#_Toc26300211)

[2.2 Instalación 3](#_Toc26300212)

[2.3 Prueba 3](#_Toc26300213)

[2.4 Reflexión 3](#_Toc26300214)

# 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

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

|  |
| --- |
| ObjectManager |
| Properties |
| Objects : ArrayList <SecurityObject> |
| Subjects: ArrayList <SecuritySubject> |
| Methods |
| +getInstance():ObjectManager |
| +getObjects() : ArrayList <SecurityObject> |
| +getSubjects(): ArrayList<SecuritySubject> |
| +createSubject(String,Level) : boolean |
| +createObject(String, Level) : boolean |
| -ExistObject() : boolean |
| -ExistSubject() : boolean |
| -findObjectByName(String): SecurityObject |
| -findSubjectByName(String): SecuritySubject |
| +read(String,String): boolean |
| +write(String,String,int): boolean |

InstructionObject es el representante de la instrucción completa “el comando” y el referencemanager es el encargado de ejecutar

|  |
| --- |
| InstructionObject |
| Properties |
|  |

### Instalación

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

### 1.3 Prueba

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

### 1.4 Conclusión

\*\* Conclusión de la parte 1

## 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