



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

TYPE-BASED DECLASSIFICATION EN DART: IMPLEMENTACIÓN Y  
ELABORACIÓN DE HERRAMIENTAS DE INFERENCIA

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN COMPUTACIÓN

MATÍAS IGNACIO MENESES CORTÉS

PROFESOR GUÍA:  
ÉRIC TANTER

MIEMBROS DE LA COMISIÓN:

SANTIAGO DE CHILE  
ABRIL 2018

RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN  
POR: MATÍAS IGNACIO MENESES CORTÉS  
FECHA: ABRIL 2018  
PROF. GUÍA: ÉRIC TANTER

TYPE-BASED DECLASSIFICATION EN DART: IMPLEMENTACIÓN Y  
ELABORACIÓN DE HERRAMIENTAS DE INFERENCIA

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.



*Una dedicatoria corta. Por ejemplo, A los creadores de U-Campus*



# Agradecimientos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



# Tabla de Contenido

<b>Introducción</b>	<b>1</b>
<b>1. Marco Teórico</b>	<b>3</b>
1.1. Information flow control . . . . .	3
1.1.1. Tipado de seguridad . . . . .	3
1.1.2. No-interferencia . . . . .	4
1.1.3. Declasificación . . . . .	4
1.2. Type-based declassification . . . . .	5
1.2.1. Sistema de tipos . . . . .	5
1.2.2. Relaxed noninterference . . . . .	6
1.3. Lenguaje Dart . . . . .	6
<b>2. Especificación del problema</b>	<b>7</b>
<b>3. Primero</b>	<b>8</b>
3.1. Hola . . . . .	8
<b>4. Segundo</b>	<b>10</b>
<b>Conclusión</b>	<b>12</b>



# Índice de Tablas

4.1. Tabla 1 . . . . .	14
------------------------	----

# Índice de Ilustraciones

4.1. Logo de la Facultad . . . . .	13
------------------------------------	----



# Introducción

La protección de la confidencialidad de la información manipulada por los programas computacionales es un problema cuya relevancia se ha incrementado en el último tiempo, a pesar de tener varias décadas de investigación. Por ejemplo, una aplicación web (o móvil) que como parte de su funcionamiento debe interactuar con servicios de terceros y por tanto debe proteger que su información sensible no se escape durante la ejecución de la aplicación a canales públicos.

Muchas de las técnicas de seguridad convencionales como *control de acceso* tienen deficiencias para proteger la confidencialidad de un programa, por ejemplo no restringen la propagación de información [1].

Formas más expresivas y efectivas de proteger la confidencialidad se basan en un análisis, ya sea estático o dinámico, sobre el código del programa, y se categorizan dentro de *language-based security* (LBS). En este trabajo, nos centraremos en la verificación estática de que un programa protege la confidencialidad de su información.

**Tipado de seguridad** Una de las técnicas más efectivas de LBS con análisis estático es *tipado de seguridad* en un *lenguaje de seguridad*. En un lenguaje de seguridad, los valores y los tipos son anotados con niveles de seguridad para clasificar la información que el programa manipula. Dichos niveles de seguridad forman una *lattice*<sup>1</sup>. Por ejemplo con la *lattice* de dos niveles de seguridad  $L \sqsubseteq H$ , se puede distinguir entre valores enteros públicos o de baja confidencialidad ( $Int_L$ ) y valores enteros privados o de alta confidencialidad ( $Int_H$ ). El sistema de tipos usa estos niveles de seguridad para prevenir que la información confidencial no fluya directa o indirectamente hacia canales públicos [1].

**No-interferencia** Formalmente, la propiedad de confidencialidad puede ser expresada como una propiedad de *no-interferencia* (*noninterference*). A grandes rasgos noninterference expresa que para dos ejecuciones realizadas por el adversario de un programa seguro, con valores confidenciales equivalentes, las salidas del programa deben ser equivalentes para el adversario. Esto caracteriza que el adversario no aprende nada sobre los valores confidenciales de un programa.

El siguiente programa ilustra el concepto de noninterference. Este muestra un método

---

<sup>1</sup>Un orden parcial, donde todo par de elementos tiene un único supremo e ínfimo

`login` para verificar la contraseña de un usuario. Se considera que el parámetro `password` es privado (y por tanto no provisto por el adversario), mientras que el parámetro `guess` es público (y lo provee el adversario).

```
String login(String password, String guess) {  
    if (password == guess) return "Login Successful";  
    else return "Login failed";  
}
```

Este programa no cumple noninterference, pues el adversario puede aprender sobre la variable confidencial `password` observando el valor de retorno del método para distintas ejecuciones.

**Declasificación** En una aplicación real y práctica deseamos que el programa anterior sea aceptado a pesar de violar la propiedad de no-interferencia, pues de otra forma no tendríamos cómo realizar la autenticación. Para solucionar este problema, los lenguajes de seguridad adicionan mecanismos para *declasificar* la información confidencial, implementados de diferentes formas [1]. Una de ellas, por ejemplo en Jif (un lenguaje de seguridad) [1] es usar un operador `declassify`, como se indica en el siguiente ejemplo, declasificando la comparación de igualdad del parámetro confidencial `password` con el parámetro público `guess`

```
String login(String password, String guess) {  
    if ( declassify (password == guess)) return "Login Successful";  
    else return "Login failed";  
}
```

Esto no corresponde a una amenaza de seguridad, debido a que el resultado de la operación de comparación es negligible con respecto al parámetro privado `password`. Sin embargo, usos arbitrarios del operador `declassify` pueden resultar en serias fugas de información, como por ejemplo `declassify(password)`.

Varios mecanismos se han explorado para controlar el uso de declasificación, y poder asegurar además una propiedad de seguridad para el programa [1]. En esta dirección, Cruz et al. [1] recientemente propusieron *type-based declassification* como un mecanismo de declasificación que conecta la abstracción de tipos con una forma controlada de declasificación, en una manera intuitiva y expresiva, proveyendo garantías formales sobre la seguridad del programa.

# Capítulo 1

## Marco Teórico

La protección de la confidencialidad de la información manipulada por los programas computacionales es un problema cuya relevancia se ha incrementado en el último tiempo, a pesar de tener varias décadas de investigación. Por ejemplo, una aplicación web (o móvil) que como parte de su funcionamiento debe interactuar con servicios de terceros y por tanto debe proteger que su información sensible no se escape durante la ejecución de la aplicación a canales públicos.

### 1.1. Information flow control

Muchas de las técnicas de seguridad convencionales como *control de acceso* tienen deficiencias para proteger la confidencialidad de un programa, por ejemplo no restringen la propagación de información [1].

Un mecanismo más expresivo para la protección de la confidencialidad e integridad de la información se denomina *information flow control*. Mientras control de acceso restringe qué datos pueden ser accedidos, *information flow control* restringe el flujo de estos datos.

#### 1.1.1. Tipado de seguridad

El análisis de *information flow control* puede ser realizado sobre el código del programa, de forma estática o dinámica. Una de las técnicas más efectivas de *information flow control* con análisis estático es *tipado de seguridad* en un *lenguaje de seguridad*. En un lenguaje de seguridad, los valores y los tipos son anotados con niveles de seguridad para clasificar la información que el programa manipula. Dichos niveles de seguridad forman una *lattice*<sup>1</sup>. Por ejemplo con la *lattice* de dos niveles de seguridad  $L \sqsubseteq H$ , se puede distinguir entre valores enteros públicos o de baja confidencialidad ( $Int_L$ ) y valores enteros privados o de alta

---

<sup>1</sup>Un orden parcial, donde todo par de elementos tiene un único supremo e ínfimo

confidencialidad ( $Int_H$ ). El sistema de tipos usa estos niveles de seguridad para prevenir que la información confidencial no fluya directa o indirectamente hacia canales públicos [].

### 1.1.2. No-interferencia

Formalmente, la propiedad de confidencialidad puede ser expresada como una propiedad de *no-interferencia* (*noninterference*). A grandes rasgos *noninterference* expresa que para dos ejecuciones realizadas por el adversario de un programa seguro, con valores confidenciales equivalentes, las salidas del programa deben ser equivalentes para el adversario. Esto caracteriza que el adversario no aprende nada sobre los valores confidenciales de un programa.

El siguiente programa ilustra el concepto de *noninterference*. Este muestra un método `login` para verificar la contraseña de un usuario. Se considera que el parámetro `password` es privado (y por tanto no provisto por el adversario), mientras que el parámetro `guess` es público (y lo provee el adversario).

```
String login(String password, String guess) {  
    if (password == guess) return "Login Successful";  
    else return "Login failed";  
}
```

Este programa no cumple *noninterference*, pues el adversario puede aprender sobre la variable confidencial `password` observando el valor de retorno del método para distintas ejecuciones.

### 1.1.3. Declasificación

En una aplicación real y práctica deseamos que el programa anterior sea aceptado a pesar de violar la propiedad de no-interferencia, pues de otra forma no tendríamos cómo realizar la autenticación. Para solucionar este problema, los lenguajes de seguridad adicionan mecanismos para *declasificar* la información confidencial, implementados de diferentes formas []. Una de ellas, por ejemplo en Jif (un lenguaje de seguridad) [] es usar un operador `declassify`, como se indica en el siguiente ejemplo, declasificando la comparación de igualdad del parámetro confidencial `password` con el parámetro público `guess`

```
String login(String password, String guess) {  
    if ( declassify (password == guess)) return "Login Successful";  
    else return "Login failed";  
}
```

Esto no corresponde a una amenaza de seguridad, debido a que el resultado de la operación de comparación es negligible con respecto al parámetro privado `password`. Sin embargo, usos arbitrarios del operador `declassify` pueden resultar en serias fugas de información, como por ejemplo `declassify(password)`.

## 1.2. Type-based declassification

Varios mecanismos se han explorado para controlar el uso de declasificación, y poder asegurar además una propiedad de seguridad para el programa []. En esta dirección, Cruz et al. [] recientemente propusieron *type-based declassification* como un mecanismo de declasificación que conecta la abstracción de tipos con una forma controlada de declasificación, en una manera intuitiva y expresiva, proveyendo garantías formales sobre la seguridad del programa.

### 1.2.1. Sistema de tipos

En *type-based declassification* los tipos tienen dos facetas, una que refleja el tipo de implementación y otro tipo que refleja las operaciones de declasificación sobre los valores de dicho tipo. Por ejemplo, el tipo `StringEq`  $\triangleq$  `[eq : String  $\rightarrow$  Bool]` autoriza la operación `eq` sobre un `String`. Entonces se puede usar el tipo de dos facetas `String < StringEq`, en donde `String` es un subtipo de `StringEq`, para controlar la operación de declasificación de la igualdad sobre `password`.

```
String login(String<StringEq password, String guess) {  
    if (password.eq(guess)) return "Login successful";  
    else return "Login failed";  
}
```

Para formalizar y demostrar propiedades del sistema de tipos, se utilizó el lenguaje `ObSEC`, que se muestra en la figura.

`ObSEC` es un lenguaje minimalista, y por tanto no soporta características comunes de lenguajes de programación, como asignaciones a variables y condicionales. La justificación de su uso se debe a que era suficiente para formular y demostrar la proposición.

### Programas bien tipados

Se demostró que programas de `ObSEC` bien tipados son *safe*<sup>2</sup>.

---

<sup>2</sup>Sin errores de tipos en tiempo de ejecución



### 1.2.2. Relaxed noninterference

La propiedad de seguridad que se demuestra para  $\text{Ob}_{\text{SEC}}$  es una forma de no-interferencia con políticas de declasificación, denominada *Relaxed noninterference*. Un lenguaje de seguridad que tiene esta propiedad, garantiza que la información confidencial solo puede fluir hacia canales públicos de una forma controlada, por medio de las políticas de declasificación.

## 1.3. Lenguaje Dart

Dart [1] es un lenguaje de programación de propósito general, orientado a objetos y de código abierto desarrollado por Google. Es usado para construir aplicaciones web, móviles y dispositivos IoT (Internet of Things).

Dart es un lenguaje de creciente adopción. Actualmente, la interfaz de Google Adwords está construida sobre Dart y Angular2. Además, Dart es usado en el framework de desarrollo multiplataforma *Flutter*.

El lenguaje Dart fue escogido porque los investigadores que realizaron el trabajo de *type-based declassification* estudian este lenguaje como parte de un proyecto de investigación mayor en el área de seguridad. Es factible implementar el enfoque de *type-based declassification* en Dart, dado que fue formalizado considerando un lenguaje minimalista orientado a objetos ( $\text{Ob}_{\text{SEC}}$ ).

## Capítulo 2

### Especificación del problema

# Capítulo 3

## Primero

### 3.1. Hola

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

**Definición 3.1** (ver [1]) *Definición definitiva*

$$\frac{d}{dx} \int_a^x f(y) dy = f(x).$$

# Capítulo 4

## Segundo

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

Maecenas dui. Aliquam volutpat auctor lorem. Cras placerat est vitae lectus. Curabitur massa lectus, rutrum euismod, dignissim ut, dapibus a, odio. Ut eros erat, vulputate ut, interdum non, porta eu, erat. Cras fermentum, felis in porta congue, velit leo facilisis odio, vitae consectetur lorem quam vitae orci. Sed ultrices, pede eu placerat auctor, ante ligula rutrum tellus, vel posuere nibh lacus nec nibh. Maecenas laoreet dolor at enim. Donec molestie dolor nec metus. Vestibulum libero. Sed quis erat. Sed tristique. Duis pede leo, fermentum quis, consectetur eget, vulputate sit amet, erat.

Donec vitae velit. Suspendisse porta fermentum mauris. Ut vel nunc non mauris pharetra varius. Duis consequat libero quis urna. Maecenas at ante. Vivamus varius, wisi sed egestas tristique, odio wisi luctus nulla, lobortis dictum dolor ligula in lacus. Vivamus aliquam, urna sed interdum porttitor, metus orci interdum odio, sit amet euismod lectus felis et leo. Praesent ac wisi. Nam suscipit vestibulum sem. Praesent eu ipsum vitae pede cursus venenatis. Duis sed odio. Vestibulum eleifend. Nulla ut massa. Proin rutrum mattis sapien. Curabitur dictum gravida ante.

Phasellus placerat vulputate quam. Maecenas at tellus. Pellentesque neque diam, dignissim ac, venenatis vitae, consequat ut, lacus. Nam nibh. Vestibulum fringilla arcu mollis arcu. Sed et turpis. Donec sem tellus, volutpat et, varius eu, commodo sed, lectus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque enim arcu, suscipit nec, tempus at, imperdiet vel, metus. Morbi volutpat purus at erat. Donec dignissim, sem id semper tempus, nibh massa eleifend turpis, sed pellentesque wisi purus sed libero. Nullam lobortis tortor vel risus. Pellentesque consequat nulla eu tellus. Donec velit. Aliquam fermentum, wisi ac rhoncus iaculis, tellus nunc malesuada orci, quis volutpat dui magna id mi. Nunc vel ante. Duis vitae lacus. Cras nec ipsum.

Morbi nunc. Aliquam consectetur varius nulla. Phasellus eros. Cras dapibus porttitor risus. Maecenas ultrices mi sed diam. Praesent gravida velit at elit vehicula porttitor. Phasellus nisl mi, sagittis ac, pulvinar id, gravida sit amet, erat. Vestibulum est. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur id sem elementum leo rutrum hendrerit. Ut at mi. Donec tincidunt faucibus massa. Sed turpis quam, sollicitudin a, hendrerit eget, pretium ut, nisl. Duis hendrerit ligula. Nunc pulvinar congue urna.

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Aenean laoreet aliquam orci. Nunc interdum elementum urna. Quisque erat. Nullam tempor neque. Maecenas velit nibh, scelerisque a, consequat ut, viverra in, enim. Duis magna. Donec odio neque, tristique et, tincidunt eu, rhoncus ac, nunc. Mauris malesuada malesuada elit. Etiam lacus mauris, pretium vel, blandit in, ultricies id, libero. Phasellus bibendum erat ut diam. In congue imperdiet lectus.

Aenean scelerisque. Fusce pretium porttitor lorem. In hac habitasse platea dictumst. Nulla sit amet nisl at sapien egestas pretium. Nunc non tellus. Vivamus aliquet. Nam adipiscing euismod dolor. Aliquam erat volutpat. Nulla ut ipsum. Quisque tincidunt auctor augue. Nunc imperdiet ipsum eget elit. Aliquam quam leo, consectetur non, ornare sit amet, tristique quis, felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque interdum quam sit amet mi. Pellentesque mauris dui, dictum a, adipiscing ac, fermentum sit amet, lorem.

Ut quis wisi. Praesent quis massa. Vivamus egestas risus eget lacus. Nunc tincidunt, risus quis bibendum facilisis, lorem purus rutrum neque, nec porta tortor urna quis orci. Aenean aliquet, libero semper volutpat luctus, pede erat lacinia augue, quis rutrum sem ipsum sit amet pede. Vestibulum aliquet, nibh sed iaculis sagittis, odio dolor blandit augue, eget mollis urna tellus id tellus. Aenean aliquet aliquam nunc. Nulla ultricies justo eget orci. Phasellus tristique fermentum leo. Sed massa metus, sagittis ut, semper ut, pharetra vel, erat. Aliquam quam turpis, egestas vel, elementum in, egestas sit amet, lorem. Duis convallis, wisi sit amet mollis molestie, libero mauris porta dui, vitae aliquam arcu turpis ac sem. Aliquam aliquet dapibus metus.

Vivamus commodo eros eleifend dui. Vestibulum in leo eu erat tristique mattis. Cras at elit. Cras pellentesque. Nullam id lacus sit amet libero aliquet hendrerit. Proin placerat, mi non elementum laoreet, eros elit tincidunt magna, a rhoncus sem arcu id odio. Nulla eget leo a

leo egestas facilisis. Curabitur quis velit. Phasellus aliquam, tortor nec ornare rhoncus, purus urna posuere velit, et commodo risus tellus quis tellus. Vivamus leo turpis, tempus sit amet, tristique vitae, laoreet quis, odio. Proin scelerisque bibendum ipsum. Etiam nisl. Praesent vel dolor. Pellentesque vel magna. Curabitur urna. Vivamus congue urna in velit. Etiam ullamcorper elementum dui. Praesent non urna. Sed placerat quam non mi. Pellentesque diam magna, ultricies eget, ultrices placerat, adipiscing rutrum, sem.

# Conclusión

Mauris ac ipsum. Duis ultrices erat ac felis. Donec dignissim luctus orci. Fusce pede odio, feugiat sit amet, aliquam eu, viverra eleifend, ipsum. Fusce arcu massa, posuere id, nonummy eu, pulvinar ut, wisi. Sed dui. Vestibulum nunc nisl, rutrum quis, pharetra eget, congue sed, dui. Donec justo neque, euismod eget, nonummy adipiscing, iaculis eu, leo. Duis lectus. Morbi pellentesque nonummy dui.

Aenean sem dolor, fermentum nec, gravida hendrerit, mattis eget, felis. Nullam non diam vitae mi lacinia consectetur. Fusce non massa eget quam luctus posuere. Aenean vulputate velit. Quisque et dolor. Donec ipsum tortor, rutrum quis, mollis eu, mollis a, pede. Donec nulla. Duis molestie. Duis lobortis commodo purus. Pellentesque vel quam. Ut congue congue risus. Sed ligula. Aenean dictum pede vitae felis. Donec sit amet nibh. Maecenas eu orci. Quisque gravida quam sed massa.

Nunc euismod, mauris luctus adipiscing pellentesque, augue ligula pellentesque lectus, vitae posuere purus velit a pede. Phasellus leo mi, egestas imperdiet, blandit non, sollicitudin pharetra, enim. Nullam faucibus tellus non enim. Sed egestas nunc eu eros. Nunc euismod venenatis urna. Phasellus ullamcorper. Vivamus varius est ac lorem. In id pede eleifend nibh consectetur faucibus. Phasellus accumsan euismod elit. Etiam vitae elit. Integer imperdiet nibh. Morbi imperdiet orci euismod mi.



Figura 4.1: Logo de la Facultad

Donec tincidunt tempor metus. Aenean egestas cursus nulla. Fusce ac metus at enim viverra lacinia. Vestibulum in magna non eros varius suscipit. Nullam cursus nibh. Mauris neque. In nunc quam, convallis vitae, posuere in, consequat sed, wisi. Phasellus bibendum consectetur massa. Curabitur quis urna. Pellentesque a justo.

In sit amet dui eget lacus rutrum accumsan. Phasellus ac metus sed massa varius auctor. Curabitur velit elit, pellentesque eget, molestie nec, congue at, pede. Maecenas quis tellus non lorem vulputate ornare. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam magna arcu, vulputate egestas, aliquet ut, facilisis ut, nisl.



Donec vulputate wisi ac dolor. Aliquam feugiat nibh id tellus. Morbi eget massa sit amet purus accumsan dictum. Aenean a lorem. Fusce semper porta sapien.

Campo 1	Campo 2
Valor 1	Valor2

Tabla 4.1: Tabla 1

Curabitur sit amet libero eget enim eleifend lacinia. Vivamus sagittis volutpat dui. Suspendisse potenti. Morbi a nibh eu augue fermentum posuere. Curabitur elit augue, porta quis, congue aliquam, rutrum non, massa. Integer mattis mollis ipsum. Sed tellus enim, mattis id, feugiat sed, eleifend in, elit. Phasellus non purus sed elit viverra rhoncus. Vestibulum id tellus vel sem imperdiet congue. Aenean in arcu. Nullam urna justo, imperdiet eget, volutpat vitae, semper eu, quam. Sed turpis dui, porttitor ut, egestas ac, condimentum non, wisi. Fusce iaculis turpis eget dui. Quisque pulvinar est pellentesque leo. Ut nulla elit, mattis vel, scelerisque vel, blandit ut, justo. Nulla feugiat risus in erat.

# Bibliografía

- [1] Ioannis Karatzas. and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, Berlin, 2nd edition, 2000.