



# 10 Pines

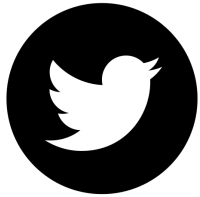
Diseño a la Gorra  
Temporada 02 - Episodio 02



Hernán Wilkinson



hernan.wilkinson@10pines.com



@HernanWilkinson



<https://alagorra.10pines.com>

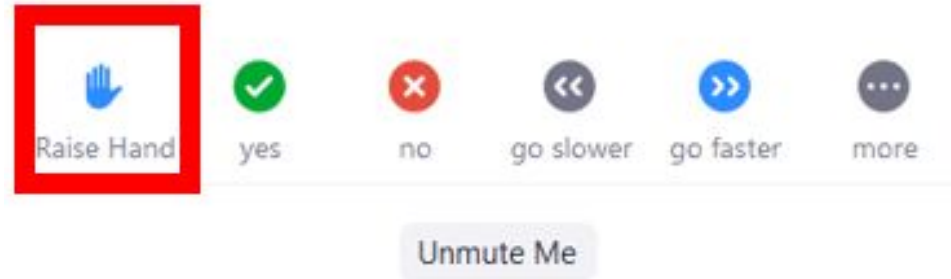


#diseño-a-la-gorra en Academia-10Pines  
(si no recibieron invitación avisen!)





Estar muteados a menos que sea necesario



No voy a poder leer el chat



La comunicación visual es importante. Usarla a discreción





# Diseño ¡a la gorra!

¡Bienvenidos a este exitoso Webinar que se llama "*Diseño a la Gorra*"!

Durante **el año 2020** hicimos **17 ediciones** donde exploramos qué es diseñar, cómo hacerlo mejor, cómo desarrollar un sistema haciendo TDD, etc.

En **el año 2022** tendremos nuevos episodios en los que tocaremos todo tipo de tema, desde diseño de lenguajes de programación y su impacto en nuestro trabajo hasta temas de diseño clásico. A todo esto le sumaremos sorteos a los cursos abiertos que estaremos dando en 10Pines.

Te esperamos todos los **Miércoles a las 19 Hrs GMT-3** a partir del Miércoles 2 de Marzo de 2022. Para poder participar [REGISTRARTE ACA](#).

Todo el código y presentaciones estarán disponibles para que lo puedan usar y consultar en cualquier momento [acá](#).

👉 Más abajo están los links a todos los videos 👉

## Pago voluntario



\$1.000 - Muy buena 🍺

Pagar

\$1.500 - Dos combos 🍔

Pagar

\$2.000 - Dos buenos 🍷

Pagar



Donate



[¿Querés abonar un monto diferente o en otra plataforma?](#)

<https://alagorra.10pines.com>

Online

# Historia y Filosofía de los Lenguajes de Programación

📅 Empieza el 29/03

📅 Del 29/03 al 21/04. Días: Todos los Martes de 19:00 a 21:00, GMT-3.

💰 AR\$ 64.800- (IVA incluido) U\$D 495- (impuestos incluidos)

Hacer una consulta

Inscribirme ahora **0% Dto**



Dictado por:  
**Hernán Wilkinson**

~~AR\$ 72.000-~~


**AR\$ 64.800-**

(IVA incluido)

~~U\$D 550-~~

**U\$D 495-**

(impuestos incluidos)

 **Early Bird** Descuento por Inscripción Temprada hasta 08/03

<http://academia.10pines.com>





Online

# Historia y Filosofía de los Lenguajes de Programación

📅 Empieza el **29/03**

📅 Del 29/03 al 21/06, 3 días todos los Martes de 19:00 a 21:00, GMT-3.

💰 **AR\$ 64.800-** (IVA incluido) **U\$D 495-** (impuestos incluidos)

Hacer una consulta

Inscribirme ahora **10% Off**



Dictado por:  
**Hernán Wilkinson**


~~AR\$ 72.800-~~  
**AR\$ 64.800-**

(IVA incluido)

~~U\$D 550-~~

**U\$D 495-**

(impuestos incluidos)

 **Early Bird** Descuento por Inscripción Temprada hasta 08/03

<http://academia.10pines.com>





## Ansiosos VS Masoquistas



¿Quién ganará hoy?



Online

# Diseño Avanzado de Software con Objetos I

📅 Empieza el **04/04**

📅 Del 04/04 al 18/04. Días: 3 días a la semana durante 2 semanas — Lunes, Miércoles y Viernes — 9:00 a 13:00 GMT-3.

💰 **AR\$ 64.800-** (IVA incluido) **USD 495-** (impuestos incluidos)

Hacer una consulta

Inscribirme ahora **10% Dto.**



Dictado por:  
**Máximo Prieto**

~~AR\$ 72.000-~~

**AR\$ 64.800-**

(IVA incluido)

~~USD 550-~~

**USD 495-**

(impuestos incluidos)

🕒 **Early Bird** Inscripción Temprana hasta 21/03

Online

# Diseño Avanzado de Software con Objetos II

📅 Empieza el **09/05**

📅 Del 09/05 al 20/05. Días: 3 días a la semana durante 2 semanas — Lunes, Miércoles y Viernes — 9:00 a 13:00 GMT-3..

💰 **AR\$ 64.800-** (IVA incluido) **USD 495-** (impuestos incluidos)

Hacer una consulta

Inscribirme ahora **10% Dto.**



Dictado por:  
**Hernán Wilkinson**

~~AR\$ 72.000-~~

**AR\$ 64.800-**

(IVA incluido)

~~USD 550-~~

**USD 495-**

(impuestos incluidos)


🕒 **Early Bird** Descuento por Inscripción Temprana hasta 18/04


<http://academia.10pines.com>



Online


# Test Driven Development Avanzado

 Empieza el **23/05**

 **AR\$ 64.800-** (IVA incluido) **USD 495-** (impuestos incluidos)


Hacer una consulta

Inscribirme ahora **10% Dto.**

 Dictado por:  
**Hernán Wilkinson**


AR\$ 72.000-


**AR\$ 64.800-**  
(IVA incluido)  
~~USD 550-~~  
**USD 495-**  
(impuestos incluidos)

 **Early Bird** Descuento por Inscripción Temprana hasta 28/04

Online


# Construcción de Software Robusto con TDD

 Empieza el **20/04**

 **AR\$ 64.800-** (IVA incluido) **USD 495-** (impuestos incluidos)


Hacer una consulta


Inscribirme ahora **10% Dto.**

 Dictado por:  
**Máximo Prieto**

AR\$ 72.000-

**AR\$ 64.800-**  
(IVA incluido)  
~~USD 550-~~  
**USD 495-**  
(impuestos incluidos)

 **Early Bird** Descuento por Inscripción Temprana hasta 04/04

 Del 20/04 al 02/05. Días: 3 días a la semana durante 2 semanas — Lunes, Miércoles y Viernes — 9:00 a 13:00 GMT-3..

<http://academia.10pines.com>

# “Errores” históricos de los Lenguajes de Programación Parte I - Repaso



# Principio del menor asombro

(Principle Of Least Astonishment)

El resultado de realizar alguna “operación” debe ser obvio, consistente y predecible, según el nombre de la “operación” y otras pistas

Si nos equivocamos, quiero saberlo  
rápidamente y claramente  
(Que la computadora nos enseñe)

<http://wiki.c2.com/?PrincipleOfLeastAstonishment>



# Principio del menor asombro

(Principle Of Least Astonishment)

Cuando no se cumple:

- Complejidad Accidental
- Confusión
- Errores...





# “Errores” o “Features”

- Estamos viendo los “errores” o “features” que generan resultados o errores inesperados
- No están relacionados con errores de lógica o algoritmia
- Generalmente no cumplen con tener en cuenta:
  - El paso del tiempo
    - Ej: dependencias temporales en el código
  - El trabajo en equipo
  - Enseñarnos cómo usar el lenguaje



# “Errores” de Parte I

- Definición automática de Variables
- Binding dinámico de Variables
  - No es lo mismo que binding dinámico de mensaje/método
- Binding de ***this*** en JavaScript
- Creación de “nameless objects”
  - Abuso de diccionarios/hashes/etc
- No chequeo de índice al acceder a un Array



# “Errores” históricos de los Lenguajes de Programación Parte II



# “Error” Nro 1: Null - The Billion Dollar Mistake





# Tony Hoare - Array bounds check in Algol 60



# Tony Hoare - “record handling”




# Tony Hoare - Ventajas de “record handling”



# Tony Hoare - Record Handling

## RECORD HANDLING

C. A. R. Hoare



### Introduction

Problems involving structured data arise in many areas of computer application, for example, in simulation studies, information retrieval, graph theory, compiler writing, symbol manipulation, and business oriented data processing. In each of these areas, specialised languages have been developed to define and manipulate data structures of a form suited to that particular application. This paper presents a generalised approach to data structures, which is suitable for incorporation in a general purpose programming language such as ALGOL 60, and which may extend its application to many of the areas cited above.

The principles which have been observed in the design of record handling are conceptual economy, security and efficiency.





# Tony Hoare - Record Handling

## 1.1 Objects and Records.

A fundamental feature of our understanding of the world is that we organise our experience as a number of distinct objects (tables and chairs, bank loans and algebraic expressions, polynomials and persons, transistors and triangles, etc); and our thought, language, and actions are based on the designation, description, and manipulation of these objects, either individually or in relationship with other objects. When we wish to solve a problem on a computer, we often need to construct within the computer a model of that aspect of the real or conceptual world to which the solution of the problem will be applied. In such a model, each object of interest must be represented by some computer quantity which is accessible to the program, and which can be manipulated by it. Such a quantity is known as a record.



# Tony Hoare - Record Handling

## 1.2 Attributes and Fields.

Each of the objects represented in the computer by a record will possess one or more attributes which are relevant to the solution of the problem. Such attributes can generally be indicated by simple values of some appropriate type. For example, the rate of interest on a bank loan can

## 1.3 Collective Nouns and Record Classes.

The objects of the real world are often conveniently classified into a number of mutually exclusive classes, and each class is named by some collective noun, such as "person", "bankloan", "expression", etc. In general, all the members of a given class have the same sort of attributes, whereas attributes which are appropriate for members of one class are totally inappropriate for members of other classes. Thus it is absurd to enquire about the rate of interest of a person, or conversely, the sex of a bankloan. In the computer model, records are also grouped into mutually exclusive record classes; each record of a given class has the same number of fields as every other record of that class, and the fields have the same types and identifiers. Records of another class will often have a different



# Tony Hoare - Record Handling

## 2.1 Functional Relationships and References.

In addition to possessing attributes, the objects of the real world may also bear certain relationships to each other. For example, one person may be the father of another person, a town may be the destination of a certain road, a subexpression may be the left operand of a given expression, etc. The programmer will sometimes want to represent such relationships

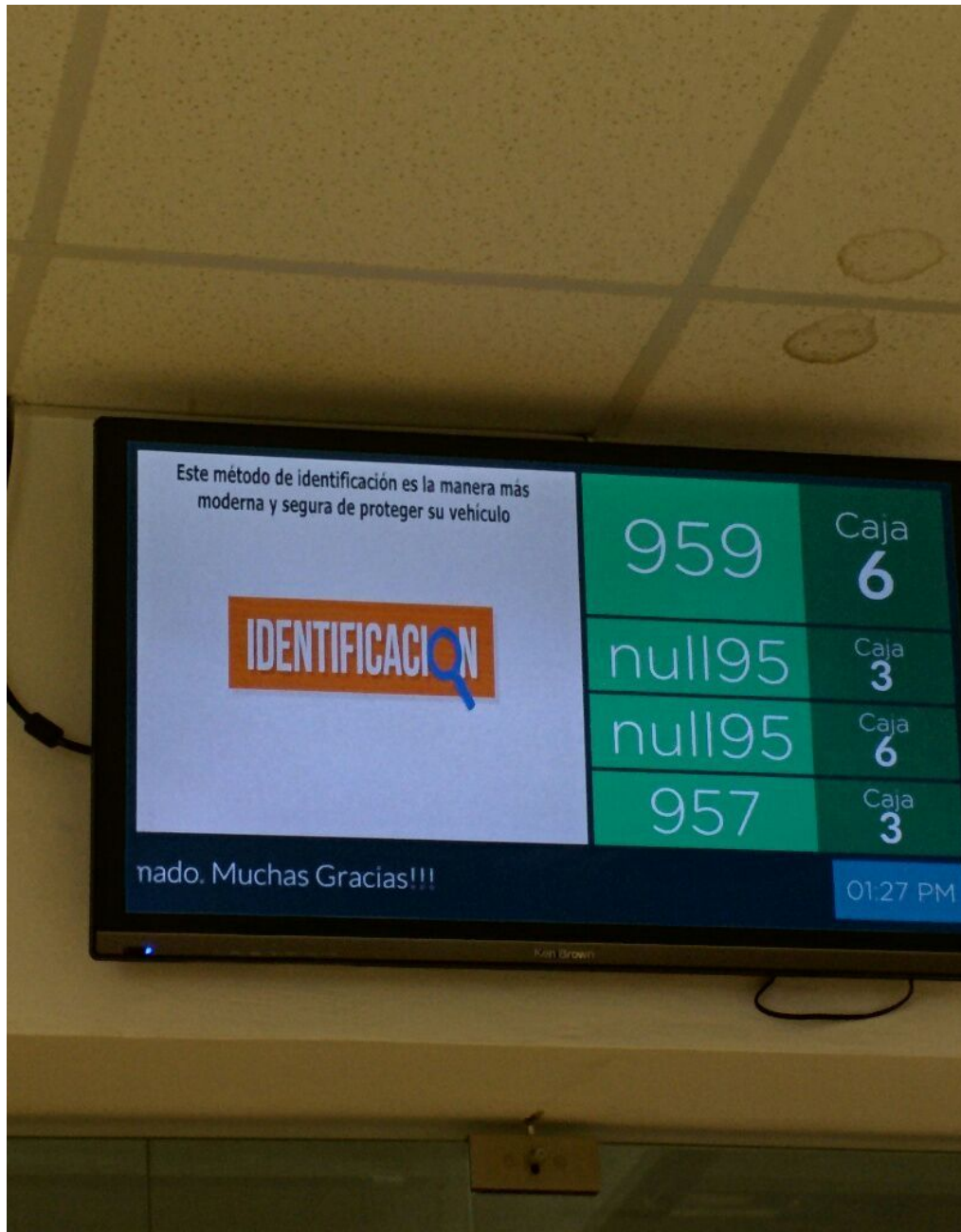


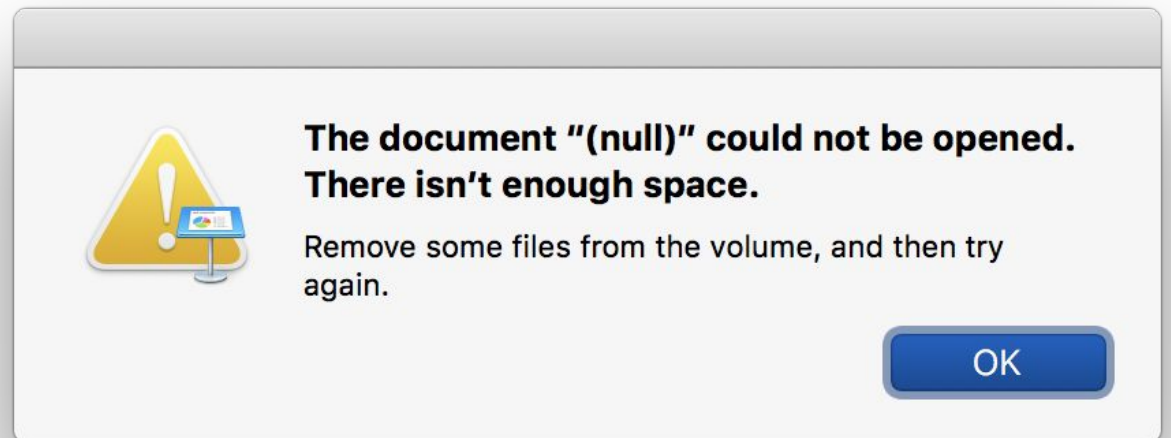
# Tony Hoare y null ...











## Promocionar publicación

Presupuesto total ⓘ

100 null

✓ 100 null

3.500 null

24.000 null

49.000 null

162.000 null

Elige otro importe

Publicar este anuncio hasta el  23/6/2017

Gastarás un promedio de **33** por día. Este anuncio estará en circulación durante **3** días, hasta el 23 de junio de 2017.

## PAGO



Al hacer clic en "Promocionar", aceptas las [Condiciones de F](#)





## Promocionar publicación

Presupuesto total ⓘ

100 null

✓ 100 null

3.500 null

24.000 null

49.000 null

162.000 null

Elige otro importe

Publicar este anuncio hasta el 23/6/2017

Gastarás un promedio de **33** por día. Este anuncio estará en circulación durante **3** días, hasta el 23 de junio de 2017.

## PAGO



Al hacer clic en "Promocionar", aceptas las Condiciones de F

2. Favorite



4: Run 5: Debug 6: TODO 9: Version Control Application Servers Terminal Java Enterprise Spring Q: Messages

Push successful: Pushed 1 commit to origin/correcciones a master (8 minutes ago)

3:1



Analisemos la preocupación de Tony Hoare y la observación de Edsger Dijkstra



# Tony Hoare - La aparición de null

## 2.2 Partial Functional Relationships.

Functional relationships may be classified as either partial or total. A total functional relationship is one which satisfies the condition that for every  $x$  in the class which defines its domain, there is exactly one  $y$  in its range to which  $x$  bears the relationship. For a partial functional relationship, there may be an  $x$  for which there is no  $y$  appropriately related to it. All the relationships introduced above for persons are in fact partial, since there will be persons who have no offspring, and persons who have no elder brother or sister; and in any finite collection of persons, there must be at least one who has no father. In order to meet this problem, a special null value is provided for reference variables and fields. If a field of a record is given this value, it usually indicates that the functional relationship represented by that field is not defined (or not yet defined) for that record; and that it is therefore a partial rather than total functional relationship. The null reference value is denoted by the basic symbol null.





# Tony Hoare - null

## 2.6 Representation of the null Reference.

The null reference value should be represented in the computer by an integer value which is outside the range of addresses of the storage area allocated for holding records. It is highly desirable that, on a computer with any form of storage protection, it should be the address of the beginning of the protected area. This would mean that if a null reference is accidentally used by a field designator at run time, the error will be immediately detected by hardware, and there is no risk of unpredictable or calamitous results.

If a computer does not possess hardware storage protection of any kind, it may still be possible to choose a value outside the address range of the computer, so that an attempt to use a null reference incorrectly cannot corrupt the store, and may even give consistent results. However, on computers with cyclic store addressing and no protected area, it would be advisable to set aside a pseudo-protected area of store, to take its address as the null reference value, and to fill the entire area with null references. This means that any field designator occurring in an expression will always give uniform results, and run-time checking can be confined to the case of assignment to field designators.



# Simula 67 - none (null)

## 3.2.2. *Semantics*

A reference expression is a rule for computing a reference value. Thereby reference is made to an object, except if the value is none, which is a reference to "no object".

i) *Qualification*. A variable or function designator is qualified according to its declaration or specification. An object designator or local reference is qualified by the stated class identifier. The expression none is not qualified.

Classes and Subclasses - Nygaard & Dahl - 1968



# ¿Soluciones?



- Optional<T> / Maybe
  - Funciona muy bien en Lenguajes Estáticamente tipados
  - No tan bien en Lenguajes Dinámicamente tipados
- Null Object
  - Sirve para ambos tipos de lenguajes
  - Hay que crear jerarquía de Existencia/No Existencia
- Encapsular null/nil y ofrecer mensajes de “control de flujo” particulares
  - Sólo válido para lenguajes dinámicamente tipados con full closures
- GenericNil??



“Error” Nro 1b:  
Generic Nil



## GenericNil

- Si se le envía un mensaje, se devuelve el mismo
- Utilizado en Objective-C
- Oculta “problemas”



“Error” Nro 2:  
Falta de closures/lambdas/etc





El concepto de “función lambda” está implementado desde 1958, en Lisp

Ejemplo



# El concepto de bloque ejecutable existe desde Algol-60 y la implementación de Dijkstra

## Procedures and Blocks

Every procedure has the properties of a block; conversely every block which according to the ALGOL text is not a procedure can be considered as a (parameterless) procedure which is only called at one place. Since our translator does this we shall use the words "procedure" and "block" interchangeably without distinction in what follows.



# Ejemplos sin Closure y con Closure



No tener closures implica:

- Tener código repetido
- No identificar que el código es también un objeto
- No poder parametrizar código



“Los closure se hicieron famosos porque Java no los tiene” - Anónimo



Recordar: Closure != Puntero a Función  
Ejemplo C



# “Error” Nro 3: Sintaxis distinta para Closure y Bloque





Lenguajes sin closures desde su concepción  
adolecen de sintaxis distinta para closures y  
bloques



Ejemplo



## Problemas:

- Inconsistencia
- Mayor complejidad accidental
- Los “bloques” no son objetos
- Nos acostumbramos a esta diferencia!



# “Error” Nro 4: Weak Typing



# Chequeo de Tipos

- ¿Cuándo se realiza?
  - Estáticamente vs. Dinámicamente
- ¿Qué se hace cuando no se respeta un tipo?
  - Fuerte vs. Débil

Cuándo / Qué	Estáticamente (Compilación)	Dinámicamente (Ejecución)
<b>Fuerte (Strong)</b>	Java; C#; Haskell	Smalltalk; Ruby
<b>Débil (Weak)</b>	C; C++	VB6; JavaScript??



Ejemplo C++ de “intercambio de identidad”

¿Qué sucede en Java/C#?

¿Qué sucede en leng. dinámicamente tipados?





Hay que tener mucho cuidado con la conversión  
explícita de tipos en lenguajes estáticamente  
tipados débiles

En C++ usar `dynamic_cast`, que es lo que Java/C#  
implementan por default



¿Qué pasa con conversión automática de tipos?  
¿Cómo impacta si es un lenguaje weak?



# “Error” Nro 5: Conversión automática de tipos



# Ejemplo C++ y JavaScript



# La conversión explícita de tipos es “peligrosa”

- Puede generar pérdida de información
- Suele ser inconsistente
- Genera resultados inesperados



# “Feature” Nro 6: Sobrecarga de Método a partir del tipo de Parámetros



# Ejemplo Java





Tener cuidado con este “feature”, puede generar comportamiento inesperado y errores de compilación difíciles de entender



# Conclusiones





Hay que mantener los lenguajes de programación  
simples y minimales



## Agenda

- Miércoles 16: “*Errores clásicos en el diseño de software y cómo evitarlos - Parte I*” - Sorteo para Diseño I
- Miércoles 23: “*Errores clásicos en el diseño de software y cómo evitarlos - Parte II*” - Sorteo para Diseño I





# Diseño ¡a la gorra!

¡Bienvenidos a este exitoso Webinar que se llama "*Diseño a la Gorra*"!

Durante el año 2020 hicimos 17 ediciones donde exploramos qué es diseñar, cómo hacerlo mejor, cómo desarrollar un sistema haciendo TDD, etc.

En el año 2022 tendremos nuevos episodios en los que tocaremos todo tipo de tema, desde diseño de lenguajes de programación y su impacto en nuestro trabajo hasta temas de diseño clásico. A todo esto le sumaremos sorteos a los cursos abiertos que estamos dando en 10Pines.

Te esperamos todos los Miércoles a las 19 Hrs GMT-3 a partir del Miércoles 2 de Marzo de 2022. Para poder participar [REGISTRARTE ACA](#).

Todo el código y presentaciones estarán disponibles para que lo puedan usar y consultar en cualquier momento [acá](#).

👉 Más abajo están los links a todos los videos 👉

## Pago voluntario



\$1.000 - Muy buena 🍺

Pagar

\$1.500 - Dos combos 🍔

Pagar

\$2.000 - Dos buenos 🍷

Pagar



Donate



[¿Querés abonar un monto diferente o en otra plataforma?](#)

<https://alagorra.10pines.com>

¡Ahora si!



# ¡Sorteo de Entradas!

- 1 entrada gratis, 1 al 50%, 2 al 25%
- Son transferibles.
- No son de uso obligatorio

Online

## Historia y Filosofía de los Lenguajes de Programación


📅 Empieza el 29/03

📅 Del 29/03 al 21/06. Días: Todos los Martes de 19:00 a 21:00, GMT-3.


💰 AR\$ 64.800- (IVA incluido)    U\$D 495- (impuestos incluidos)

Hacer una consulta

Inscribirme ahora **10% Dto.**

 Dictado por: **Hernán Wilkinson**

~~AR\$ 72.000-~~  
**AR\$ 64.800-**  
(IVA incluido)  
~~U\$D 550-~~  
**U\$D 495-**  
(impuestos incluidos)

 **Early Bird** Descuento por Inscripción Temprada hasta 08/03





Muchas gracias





# 10 Pines

Creative Software Development



[10pines.com](http://10pines.com)



[info@10pines.com](mailto:info@10pines.com)



+54 (011) 6091-3125 / 4893-2057



Av. Leandro N. Alem 896 6° - Bs. As. - Argentina



@10pines