

# Arquitecturas de Aplicaciones

Tecnologías de Mercado

Administración de Recursos

Ing. en Sistemas de Información

FRBA – UTN – Argentina - 2014

# Características de Selección

- Nuestro análisis va a estar basado en una **visión gerencial**. Orientado a la toma de decisiones a la hora de elegir una tecnología y no desde una visión técnica.
- No nos basaremos en complejidad de programación o velocidad sino en factores tales como: **recursos humanos, costos licencias y estrategia a largo plazo**.
- Como vimos la clase pasada, el grueso de los sistemas que se desarrollan en nuestro país son del tipo **Sistemas de Gestión**. Por lo tanto vamos a centrar nuestro esfuerzo en **elegir tecnologías para estos sistemas**.
- En nuestro rol de ingenieros debemos proveer soluciones no solo basadas en el desarrollo de software sino orientadas a la resolución de problemas reales de información.
- La visión del Ingeniero es la de resolver problemas y no solo desarrollar software.

# Importancia del Conocimiento

- ¿Por qué tenemos que saber estos aspectos técnicos?
  - Todos los sistemas en la actualidad están asociados a un Software.
  - Desde una visión gerencial importa el lenguaje de desarrollo solo en términos de recursos (RRHH, tiempos, costos, etc.) y resultados
  - Si al comienzo de un proyecto se selecciona mal la tecnología dicho proyecto será mas costoso y hasta podría llegar a cancelarse
  - Desde el punto de vista gerencial no es necesario que se conozca en detalle los aspectos técnicos de los lenguajes, pero tienen que tener una visión general.
  - El tener un conocimiento actualizado de las tecnologías de mercado podría resultar en malas decisiones o en planificaciones incorrectas (estimar menos tiempos por no conocer la tecnología)

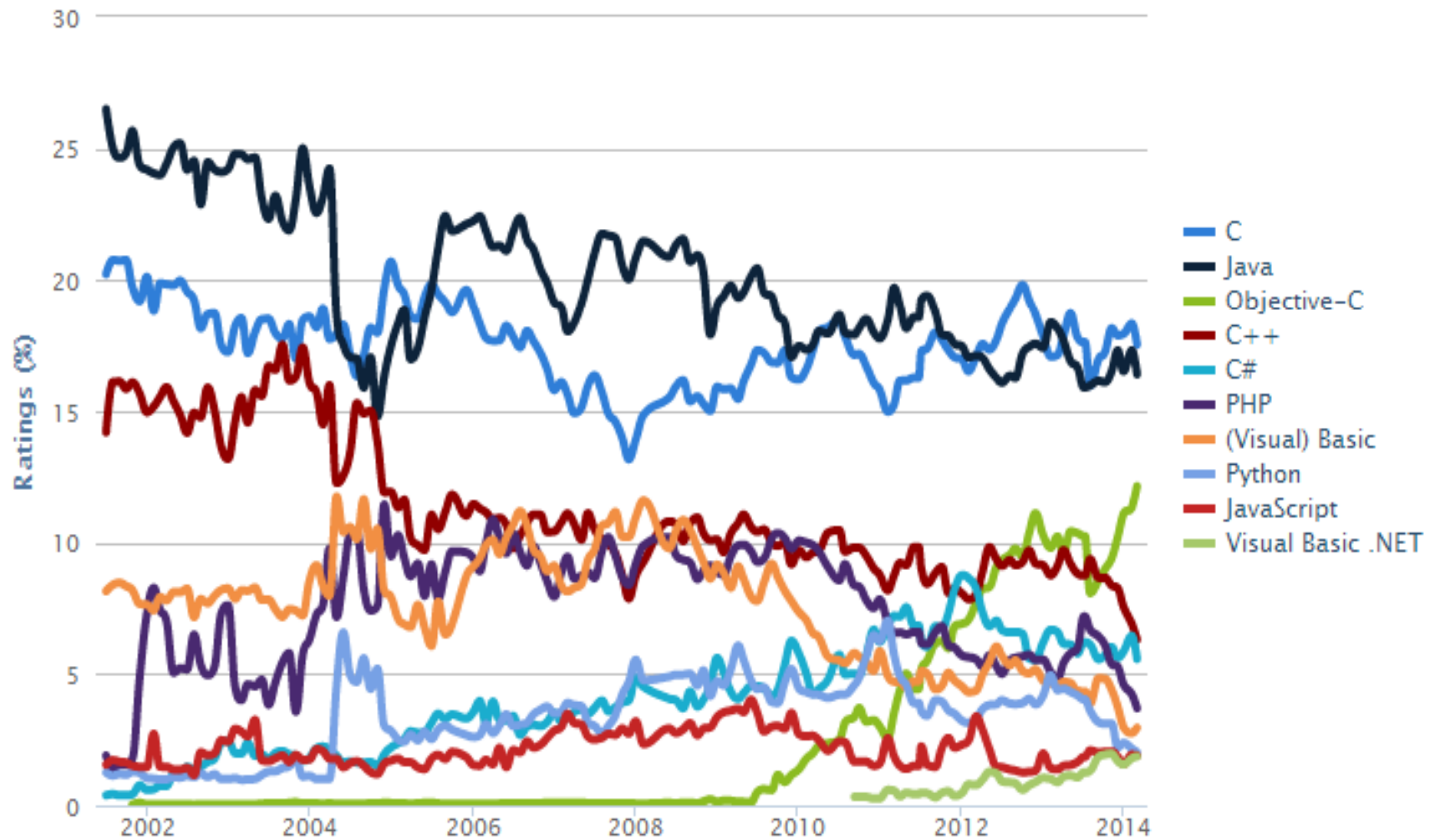
# Ranking de popularidad

Mar 2014	Mar 2013	Change	Programming Language	Ratings	Change
1	2	⬆	C	17.535%	+0.39%
2	1	⬇	Java	16.406%	-1.75%
3	3		Objective-C	12.143%	+1.91%
4	4		C++	6.313%	-2.80%
5	5		C#	5.572%	-1.02%
6	6		PHP	3.698%	-1.11%
7	7		(Visual) Basic	2.955%	-1.65%
8	8		Python	2.021%	-2.37%
9	11	⬆	JavaScript	1.899%	+0.53%
10	16	⬆	Visual Basic .NET	1.862%	+0.97%
11	17	⬆	Transact-SQL	1.477%	+0.64%
12	69	⬆	F#	1.216%	+1.14%
13	10	⬇	Perl	1.149%	-0.81%
14	9	⬇	Ruby	0.974%	-1.18%
15	15		Delphi/Object Pascal	0.881%	-0.01%
16	13	⬇	Lisp	0.747%	-0.19%
17	30	⬆	D	0.744%	+0.40%
18	20	⬆	Assembly	0.732%	+0.10%
19	14	⬇	PL/SQL	0.713%	-0.21%
20	23	⬆	MATLAB	0.688%	+0.07%

# Ranking de popularidad

TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



# Lenguaje Java Características

- Semi-interpretado, orientado a objetos, que corre sobre una maquina virtual.
- Simple de aprender a usar, ya que es intuitivo y tiene sintaxis similar a la de C.
- Al ser orientado a objetos alienta a utilizar buenas prácticas de programación, sin embargo no nos limita a hacerlo
- Es un lenguaje multiplataforma y al correr sobre una maquina virtual permite de forma relativamente sencilla portar las aplicaciones a otras arquitecturas.
- Existen versiones de Java para la mayoría de las arquitecturas (windows linux solaris mainframes).

# Lenguaje Java Ventajas

- Java es gratuito y puede armarse toda la infraestructura de desarrollo con productos gratuitos :
  - Linux en los servidores
  - Apache / Tomcat como servidor de aplicaciones.
  - Linux en la Pc de los programadores
  - Eclipse o netbeans como IDE de desarrollo
- Además como es muy utilizado mundialmente hay mucha comunidad, encontramos librerías gratuitas, foros, tutoriales y ayudas varias.
- Es una tecnología ampliamente probada.

# Lenguaje Java Desventajas

- Es Open source, por lo cual no existe soporte licenciado, por lo que puede generar desconfianza en las empresas.
- Es un lenguaje de propósito general, que no está específicamente pensado para que un desarrollo sea rápido.
- Se pueden lograr cosas muy personalizadas y es ideal para aplicaciones fuera de lo común, pero no viene listo para ser productivo y eficiente en forma inmediata.
- Debemos armarnos un framework, recolectar varias apis, etc. Por lo cual contamos con un tiempo de startup para poder comenzar a trabajar.



# .NET

- .Net, no es un lenguaje de programación, es una solución entera, incluye un lenguaje IDE, editor visual, servidor web etc.
- Esta desarrollado para utilizar la solución completa.
- Este paquete viene listo para empezar a desarrollar, lo que facilita la productividad. Se instala el paquete y ya tenemos todo configurado para comenzar a trabajar.
- Se tiene una forma de encarar el proyecto por parte de Microsoft, que no tiene porque ser la mejor ni la más óptima pero ya esta hecha, probada y funciona.

# .NET Ventajas

- Soporte licenciado
- Comienzo de desarrollo inmediato.
- Desarrollo más rápido que en otras plataformas.
- Aplicaciones integradas en el Framework Microsoft.
- Ampliamente probado.

## .NET Desventajas

- Adaptabilidad a otros productos (por ej. Google Maps o Google Docs.)
- La oferta de programadores capacitados está en crecimiento pero aun bastante más chica que la comunidad java.
- Licenciamiento, el 90% de las herramientas utilizadas son pagas.

# Lenguaje PHP

- Es open source al igual que java
- Simplicidad y facilidad de uso.
- Para un diseñador web, php es el lenguaje mas simple de aprender
- Es un lenguaje 100% interpretado, no hay una compilación intermedia como en java.
- El hecho de ser interpretado puro podría parecer que es lento, pero a diferencia de JavaScript, Php fue pensado de entrada para interpretarse lo mas rápido posible.

# Lenguaje PHP Ventajas

- Es open source.
- Simplicidad y facilidad de uso.
- Comienzo de programación inmediato.
- No se requieren deploys o compilaciones previas.
- Su Application Server Apache es muy confiable.
- Posee arquitectura de procesos en lugar de threads, de allí su confiabilidad.

# Lenguaje PHP Desventajas

- Genera mucho procesamiento al Sistema Operativo.
- El código muchas veces esta diseminado por todos los formularios.
- Menor robustez.
- Soporte limitado.

# Arquitectura de Procesos (Apache)

- Hay un proceso principal que es el propio servidor
- Por cada pedido web o request se crea un proceso para atenderlo. Este proceso atiende el pedido procesa su lógica, accede a la base de datos , genera una salida y luego muere
- Su tiempo de ejecución suele ser muy corto, del orden de los milisegundos
- En un servidor apache se crean y se mueren procesos constantemente
- El principal problema de esto es que crear un nuevo proceso requiere intervención del sistema operativo y tiene un costo asociado
- Hay que crear un montón de estructuras internas en el Sistema Operativo solo para que el proceso dure una fracción de segundo

## Arquitectura de Threads (Java -Tomcat)

- Surgen para liberar de trabajo al Sistema Operativo.
- Un threads es como un proceso pero mas liviano
- Ejecuta dentro del área de memoria de un proceso ya existente
- Los threads comparten memoria.
- Crear y matar hilos es mucho mas liviano que matar procesos
- Solo hay un proceso en el Application Server, todos los procesos se atienden creando hilos de este proceso.
- Si un thread no limpia la memoria utilizada sigue quedando ocupada, produciendo un memory leak.
- La arquitectura de servidor de threads se adapta mucho mejor a la naturaleza de las aplicaciones Web, pero requiere una mejor programación.



# Lenguaje SAP

- SAP es un producto enlatado genérico que consiste en un modelo armado y probado.
- Tiene varios módulos ya hechos, ya programados que cubren las necesidades mas comunes de una gran empresa:
  - Recursos humanos
  - Liquidación de sueldos
  - Contabilidad
  - Facturación
  - Ciclo de vida de un producto
  - Cadena de suministros

# SAP Ventajas

- SAP es un producto ya hecho y probado.
- Es ampliamente utilizado por grandes empresas en todo el mundo.
- No requiere tiempos de desarrollo previos.
- Permite su parametrización y customización.
- Ya viene con una forma de encarar la solución al problema.
- Tiene un amplio soporte.

# SAP Desventajas

- La customización que puede realizarse es limitada.
- El costo del producto es muy alto.
- No hay gran oferta de programadores ABAP, lo cual aumenta el costo de su customización.
- Los tiempos de implementación del productos son altísimos.

# Interfaces de Usuario

- Es fundamental en el desarrollo de software.
- Es la forma en que el sistema muestra la información.
- Es lo único que el usuario puede ver.
- Interfaz gráfica Desktop
- Interfaz gráfica Web

# Interfaces gráficas Desktop

- Si vamos a utilizar tecnologías Desktop tenemos las siguientes alternativas más conocidas:
  - Java
  - .NET
  - C++

# Interfaces gráficas Web

- El esquema tradicional es por Refresh o Request Response. Es el original de la Web. El usuario recibe una pantalla en Html, completa datos y la información se envía al Application Server.
- Ajax
- Tecnologías RIA:
  - Aplicaciones Ria nativas
    - Ejemplo GMAIL
    - Framework GWT
  - Aplicaciones Ria por Plugin
    - Flex
    - Silverlight