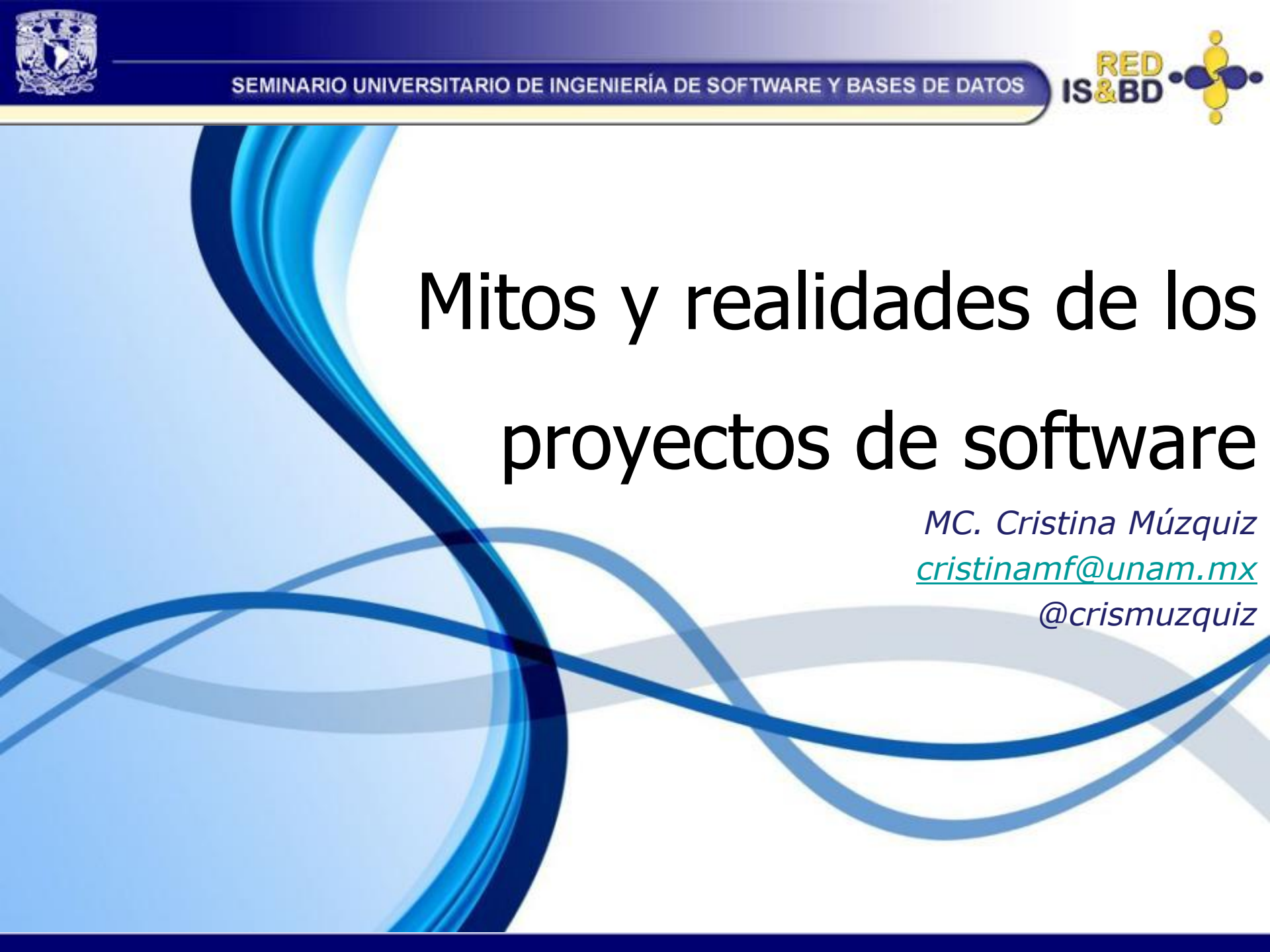


Mitos y realidades de los proyectos de software

MC. Cristina Múzquiz
cristinamf@unam.mx
@crismuzquiz

Las fases en el desarrollo de nuestros sistemas

Administración del proyecto

En el análisis de requerimientos

Mitos

En el análisis de requerimientos

La documentación no es de valor

Cualquiera puede ser analista

El cliente revisará lo que nosotros hagamos

Analizar
Estudiar
Volver a experimentar.

Cualquiera puede ser buen analista

El analista deberá ser capaz de:

- Saber que el “cliente” usualmente **no es una sola persona**, son varias, con **distintas expectativas** del proyecto.
- Aterrizar la idea general del cliente a una definición clara y viable.
- Poder **ofrecer escenarios posibles**, dentro de las limitaciones tecnológicas y de tiempo.
- Apoyar a que los interesados del proyecto entiendan cuál será el alcance y limitaciones del proyecto **(alinear expectativas).**
- **Respetar** sus conocimientos del negocio y demostrar nuestro genuino interés por ayudar.

Mito

La documentación no es de valor
Y el cliente revisará lo que hagamos

- Es cierto que aún sigue sin tener tanta importancia para nuestros clientes como el código, pero se están sensibilizando de la importancia de la documentación de los requerimientos y del proyecto.
- Estamos trabajando para que los procesos de validación sean más ligeros y en consecuencia sí nos revisen el trabajo.

Solo se programará los requerimientos que estén documentados

En el desarrollo de software

Mitos

En el desarrollo de software

Todos los programadores son iguales

Es fácil encontrar desarrolladores y es barato

Entre más desarrolladores existan, el proyecto saldrá más rápido

Teniendo los procesos bien definidos y documentados, el equipo ya sabrá que hacer y lo hará bien

Mito

Entre más desarrolladores existan, el proyecto saldrá más rápido (1/2)

A veces se parte del supuesto que:

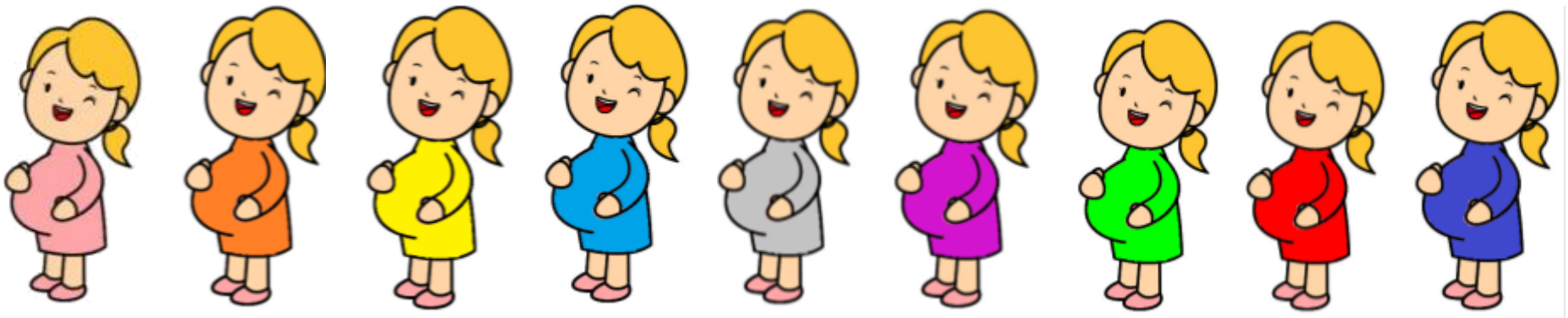
Un mes desarrollo de un programador =

Una quincena de dos desarrolladores =

Una semana de cuatro desarrolladores

Lo cual no necesariamente es cierto y sería similar a pensar que:

Nueve mamás tienen a un niño en un mes



El **Mítico Hombre-Mes: Ensayos de ingeniería de Software** (en inglés *The Mythical Man-Month: Essays on Software Engineering*) ,Fred Brooks, 1975.

Mito

Entre más desarrolladores existan, el proyecto saldrá más rápido (2/2)

El desarrollo de software no es un proceso mecánico ni lineal, cuando se incorpora un nuevo desarrollador en un proyecto que ya inició, requiere la explicación de:

- Los objetivos del proyecto.
- Los requerimientos, la arquitectura del proyecto.
- Los estándares de programación del proyecto.
- Las reglas del juego de ese grupo en particular.
- Se aumentan las líneas de comunicación n(n-1)/2

"Adding more people to a late project makes it even later!". Fred Brooks, líder del proyecto OS/360 de IBM

(Ocho personas en un equipo son: **28** líneas de comunicación, diez personas son **45**)

Teniendo los procesos bien definidos y documentados, el equipo ya sabrá que hacer y lo hará bien.

Si bien los procesos son de mucha utilidad, en todos proyecto habrá que hacer las siguientes preguntas:

- ¿Se están usando los procesos?
- ¿Todos los colaboradores los conocen y entienden?
- ¿Refleja las prácticas modernas en desarrollo de software?
- ¿Están completos?

Mito

Todos los programadores son iguales

Los programadores son personas y todas las personas son diferentes, dentro de los estilos de programadores que hemos ubicado están:

El código funciona ¿no?

No le gustan los estándares

El perfeccionista en código

No le gustan los plazos

Al que le gusta que se vea bonito

No le gustan lo que se ve “feo”

Al que le gustan los retos

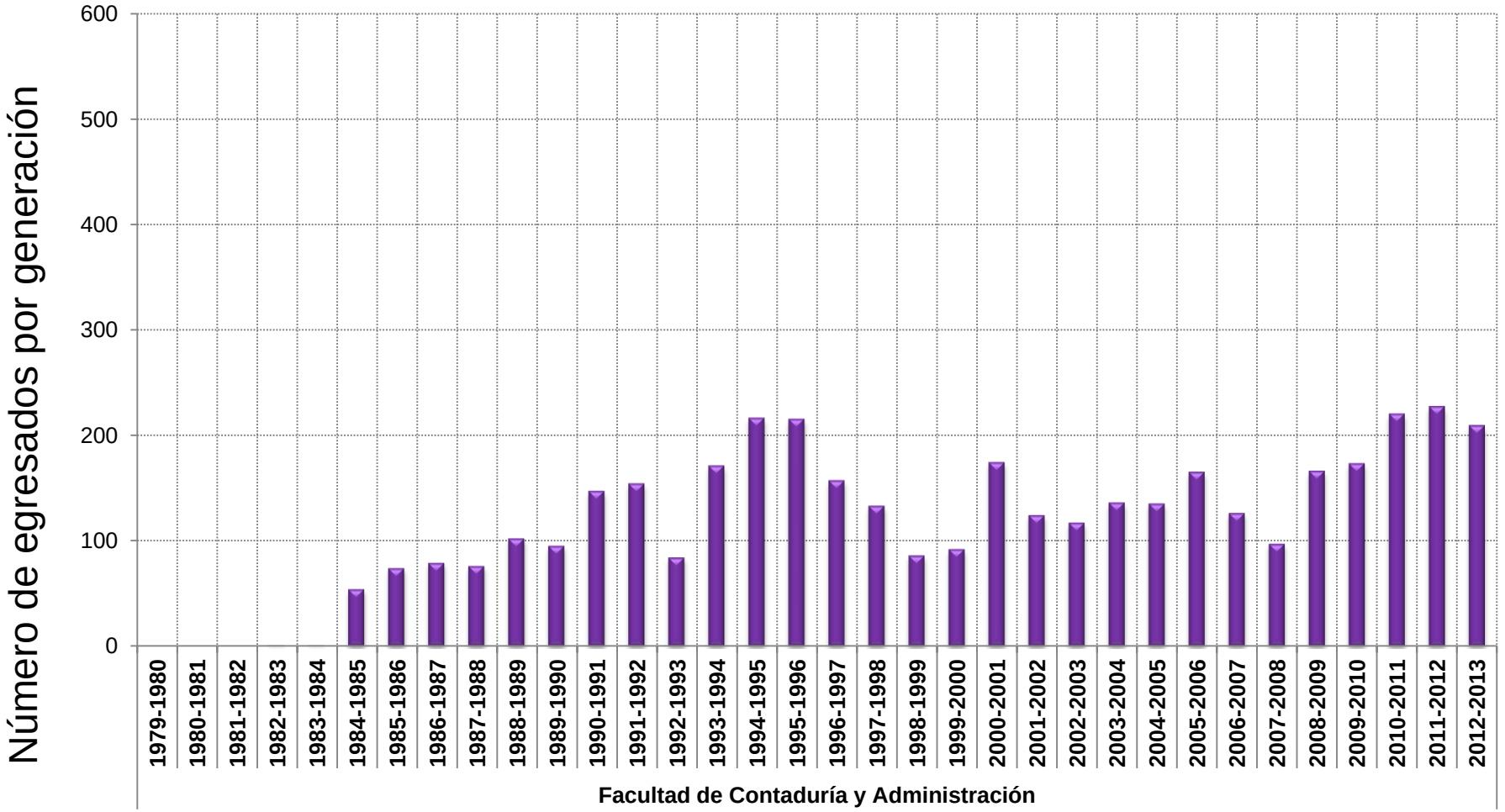
No le gusta hacer lo mismo

En nuestra experiencia a un desarrollador principiante le lleva 5 veces una actividad, en comparación con un experto.

Mito

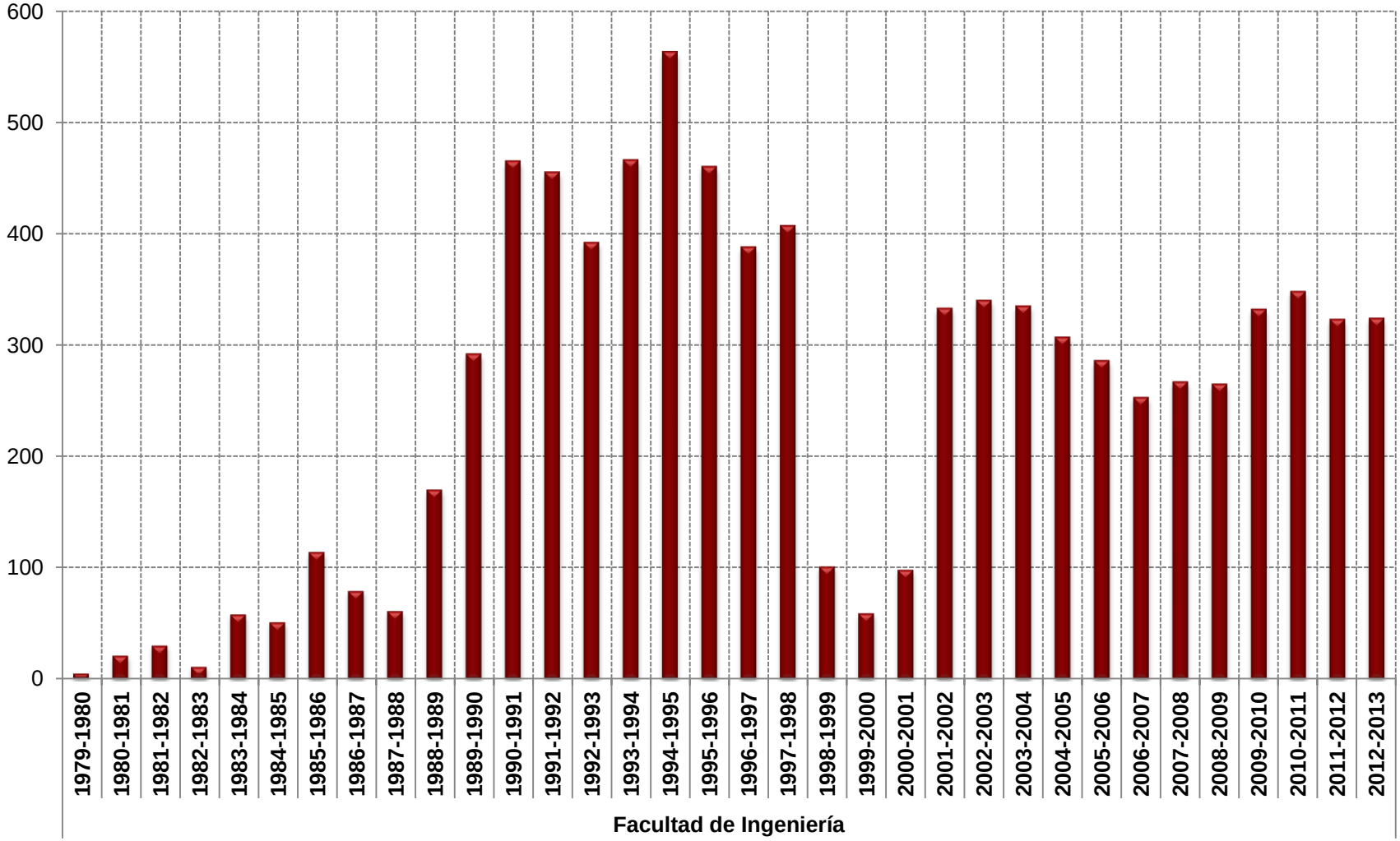
Es fácil encontrar desarrolladores y es barato

Informática



Fuente: SIDEUMS. Sistema Dinámico de Estadísticas Universitarias de la UNAM

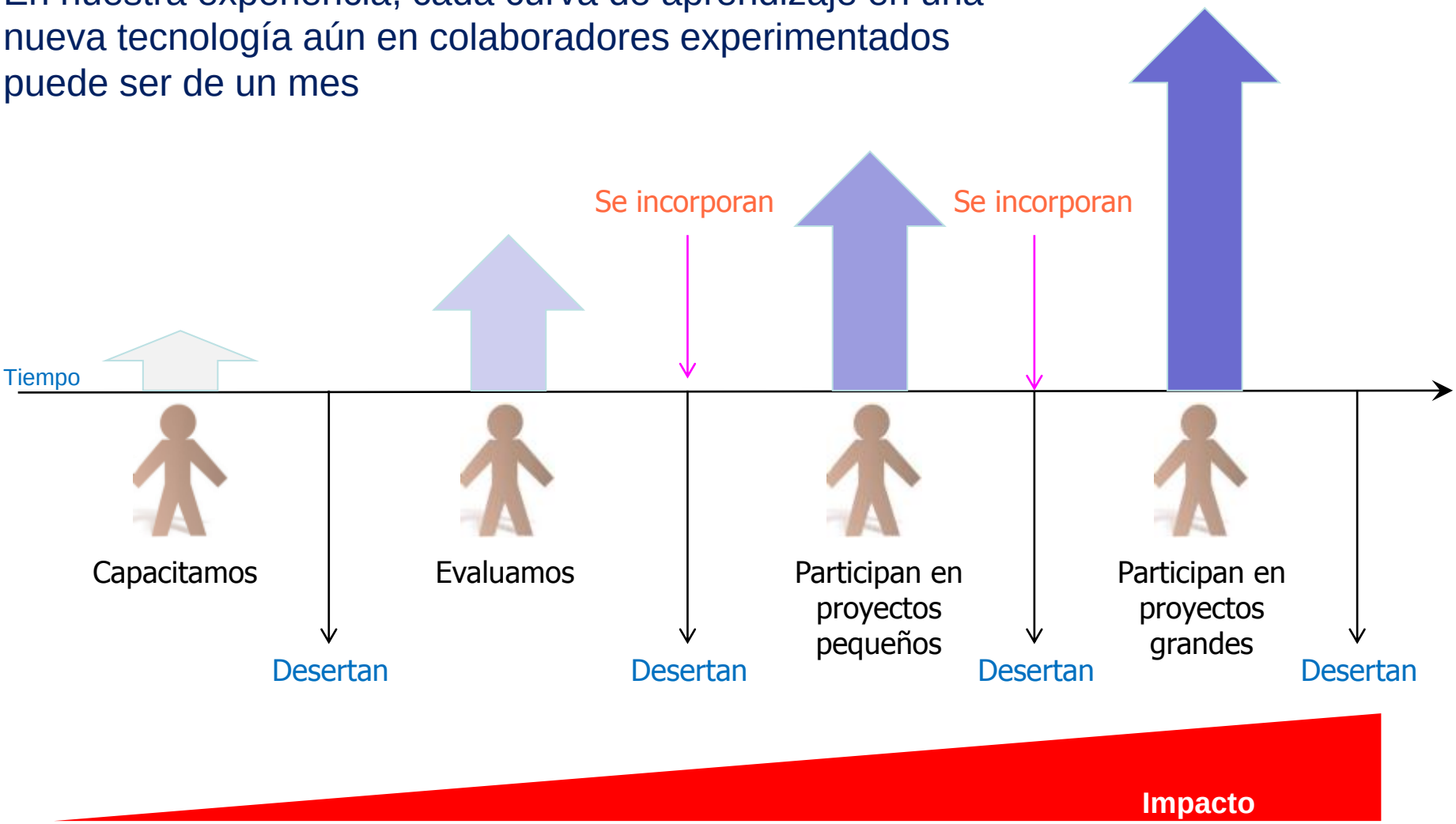
Ingeniería en Computación



Fuente: SIDEUMS. Sistema Dinámico de Estadísticas Universitarias de la UNAM

Curva de aprendizaje

En nuestra experiencia, cada curva de aprendizaje en una nueva tecnología aún en colaboradores experimentados puede ser de un mes

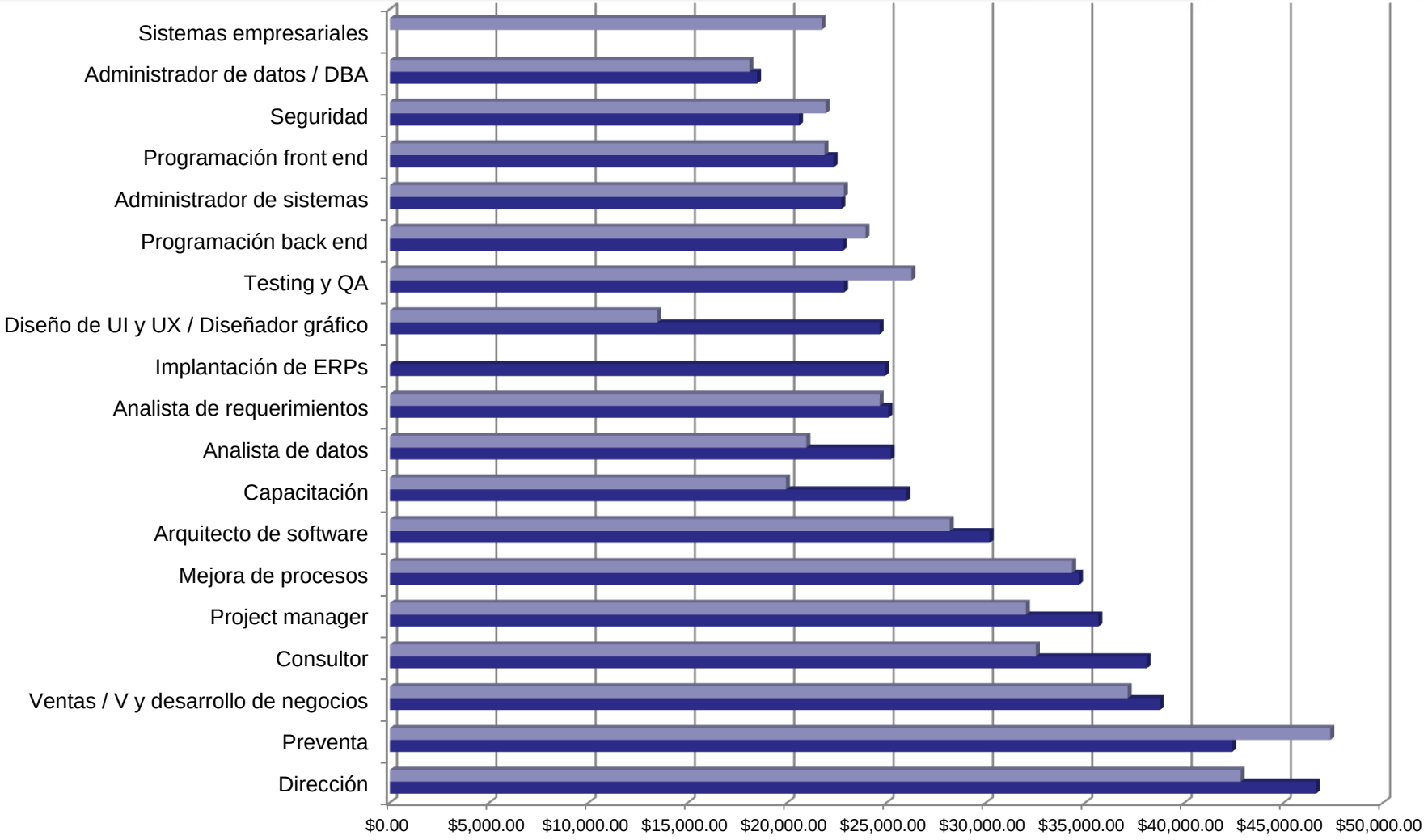


Rol	Pct	Mediana	Media	Des. Std
Dirección	4.2%	$48,500	$58,498	$38,269
Preventa	1.3%	$40,637	$50,250	$33,686
Venta	1.9%	$40,000	$53,390	$39,667
Consultoría de negocio	3.2%	$40,000	$49,855	$34,774
Project Mgmt	9.1%	$35,000	$39,345	$23,628
Mejora de procesos	3.6%	$32,500	$37,375	$25,997
Arquitectura de sistemas	9.5%	$31,000	$34,289	$22,593
Implantación de ERP	2.7%	$30,000	$33,815	$24,742
Análisis de requerimientos	10.2%	$26,000	$28,431	$17,668
Business Intelligence	2.1%	$25,000	$27,962	$15,575
Capacitación	2.4%	$25,000	$27,442	$17,027
Seguridad informática	1.0%	$25,000	$25,424	$13,525
Administración de infraestructura	3.9%	$23,000	$25,934	$15,626
Programación de Back End	18.7%	$23,000	$24,980	$15,445
Aseguramiento de la calidad	3.6%	$22,000	$25,571	$21,908
Programación de Front End	11.1%	$22,000	$24,880	$17,806
User experience design	1.4%	$21,500	$22,235	$12,487
Administración de datos	5.4%	$19,800	$22,802	$16,251
Docencia	0.8%	$16,000	$21,205	$17,677
Soporte técnico	3.6%	$15,000	$18,710	$14,273

Fuente:
Estudio de Salarios 2014. SG Buzz, Pedro Galván.
http://sg.com.mx/revista/46/estudio-salarios-2014#.VODI__mUeao
Fecha de consulta: 15/02/2015

Lenguaje	Muestra	Mediana	Media	Des. Std.
Objective C	21	$33,000	$40,141	$29,611
C	52	$25,725	$24,989	$13,762
Java	321	$25,000	$26,227	$16,342
VB	103	$25,000	$23,922	$11,809
C#	293	$24,000	$24,637	$13,549
Node JS	30	$23,970	$28,952	$20,037
PL/SQL	226	$23,450	$25,379	$16,436
Ruby	58	$23,250	$26,796	$19,870
Javascript	429	$22,700	$24,463	$14,372
Bash	47	$21,500	$24,959	$11,466
Python	40	$20,000	$23,910	$19,481
PHP	167	$18,000	$20,074	$14,016
Delphi	18	$18,000	$20,028	$9,681

Fuente: Estudio de Salarios 2014. SG Buzz, Pedro Galván. http://sg.com.mx/revista/46/estudio-salarios-2014#.VODI__mUeao
Fecha de consulta: 15/02/2015



Cifras en pesos mexicanos, sueldo bruto mensual.

Fuente: Material de Liliana Rangel, noviembre 2014, curso pruebas de software.

Pruebas de software

Mitos

En pruebas de software

Cualquiera puede ser probador

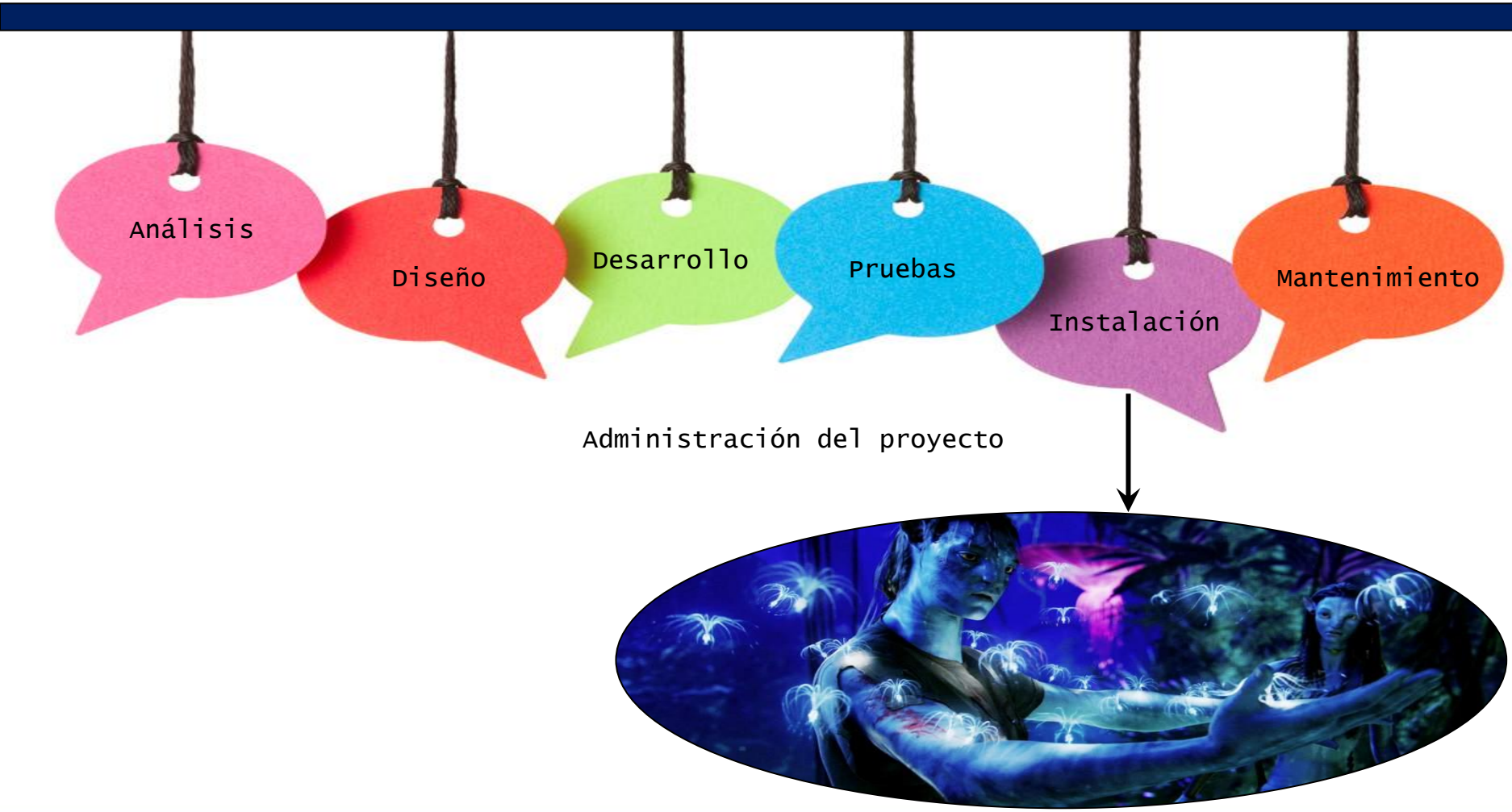
Testers certificados

- En México hay **552** certificados en "*Foundation*"
- En México hay **25** certificados en "*Advanced*"
- En nuestra área de pruebas tenemos a **3** personas certificadas en "*Foundation*" y **2** de ellas están buscando la certificación "*Advanced*".

ISTQB International Software Testing Qualifications Board

Fuente: Material de Liliana Rangel, noviembre 2014, curso pruebas de software.

Las fases en el desarrollo de nuestros sistemas



Análisis de requerimientos

Había una vez…

- Influíamos fuertemente en los requerimientos (imaginando lo que el cliente necesitaba)
- Había poca documentación
- No había un repositorio central de archivos

Hace no mucho…

- Documentamos todo en los **casos de uso**, haciendo documentos **muy densos** para leerse y validarse
- Se hacían algunos **prototipos** con **power point**
- Empezamos a promover tener una agenda conjunta de reuniones

El día de hoy…

- Estamos buscando hacer **casos de uso más ligeros** y el resto de la información la ponemos en anexos.
- Hacemos uso importante de los prototipos con **Justinmind Prototyper**
- Desde el inicio del proyecto se programan las reuniones.
- Recurrimos a otros medios no solamente de forma presencial.

Hoy, obtener la validación es de lo más importante.

Diseño

Había una vez…

- No existía la concepción del diseño del desarrollo de software en nuestros proyectos
- El enfoque era solamente en el DER y el diccionario de datos

Hace no mucho…

- Empezamos a ver la necesidad de trabajar más en la arquitectura del software, no solamente enfocada en los datos, a raíz de que se tuvieron problemas de desempeño, y de estructura del sw.
- Se tenia un procedimiento.

El día de hoy…

- El equipo de trabajo está valorando la arquitectura del sw.
- Se hace un análisis más profundo de los requerimientos del cliente para determinar la arquitectura del sistema.
- Se buscan funcionalidades en común, para hacer componentes generales.
- Hacemos uso de (Talend).

Desarrollo de software

Había una vez…

- El código se guardaba en la máquina de cada programador (Versiones no controladas)
- La construcción iniciaba antes de tener los requerimientos mínimos (Desarrollo basado en supuestos).
- Dependíamos mucho de la experiencia de las personas y cada persona tenía una forma diferente de abordar proyectos (Falta de procesos).

Hace no mucho…

- Utilizamos VSS 6.0 como repositorio de código
- Se dejaba que cada programador leyera la documentación correspondiente a sus asignaciones (Problemas de comunicación)
- Empezamos a usar estándares de programación
- Adoptamos el rol de líder técnico

El día de hoy…

- Cada proyecto tiene al menos tres ambientes (desarrollo, pruebas, pre producción)
- Utilizamos frameworks
- Controlamos las versiones del código con Subversion
- guía de instalación
- Capacitamos colectivamente al equipo de desarrollo respecto a los requerimientos
- Iniciamos con pruebas unitarias, análisis estático de código e integración continua
- Promovemos las revisiones entre pares y la comunicación entre desarrolladores
- Desarrollamos primero los componentes comunes que serán utilizados por el resto del equipo.

Hoy, hacer software de calidad es el objetivo

Lo que se detecta en una auditoría de código

Clases con excesivo número de métodos públicos

Clases con muchos métodos

Métodos con complejidad ciclomática

Métodos con complejidad Npath

Líneas de código demasiado largas

Métodos excesivamente grandes

Líneas de código repetido

Clases con complejidad excesiva

Clases excesivamente grandes

Líneas de código con errores en la identación

Errores en la nomenclatura de métodos

¿Con qué herramientas lo hacemos?

- **phploc / PHP Lines of Code**. Esta herramienta analiza el código para determinar el tamaño y la estructura general del mismo.
- **phpmd / PHP Mess Detector.** Esta herramienta identifica cuando el código tiene una **complejidad excesiva**, así mismo si hay código sin uso, la limpieza del mismo y la limpieza de la sintaxis.
- **phpcs / PHP CodeSniffer.** Esta herramienta detecta las **violaciones con respecto al estándar de codificación**, en general hace uso de los estándares usados por PEAR.
- **phpcpd / PHP Copy/Paste Detector.** Esta herramienta determina el porcentaje de **código fuente repetido** en diferentes secciones, el resultado es el porcentaje del código fuente que ha sido duplicado así como las líneas donde este se encuentra.

Ejemplo

	Controladores	Modelos	Vistas
Líneas de Código	135,893	71,233	17,3200
Clases	163	298	7
Métodos	1,347	943	16
Funciones	0	0	21

Pruebas

Había una vez...

- Pruebas consistía en hacer “maldades” al código y ver que salieran errores.

Hace no mucho...

- Poníamos el código en un repositorio como el VSS.
- Se dejaba que cada programador leyera la documentación que le correspondía a sus asignaciones.
- Hacíamos uso de estándares de programación.

El día de hoy...

- Buscamos la especialización de nuestro personal en pruebas.
- Probamos bajo buenas prácticas y bajo un marco de referencia.
- Hacemos uso de la herramienta para el manejo de las incidencias.
- Probamos en varios navegadores.
- Hacemos pruebas de desempeño y concurrencia.
- Tenemos el rol de líder de pruebas.

Hoy, no tener errores funcionales en garantía es la meta

Mantis

mantis BUG TRACKER

Conectado como: *cristinamf* (María Cristina Muzquiz Fragoso - manager) *2015-02-16 15:04 CST* Proyecto: CNMHI Cambiar

Principal | Mi Vista | Ver Incidencias | Reportar Incidencia | Log de cambios | Roadmap | Resumen | Administración | Mi Cuenta | Cerrar Sesión

Incidencia # Ir a Incidencia

Mostrando Incidencias (1 - 50 / 728) [Imprimir Informes] [Exportar a CSV] [Exportar a Excel] [Exportar XML] [Gráfico]

[Primero Anterior 1 2 3 4 5 6 7 8 9 10 11 ... Siguiente Último]

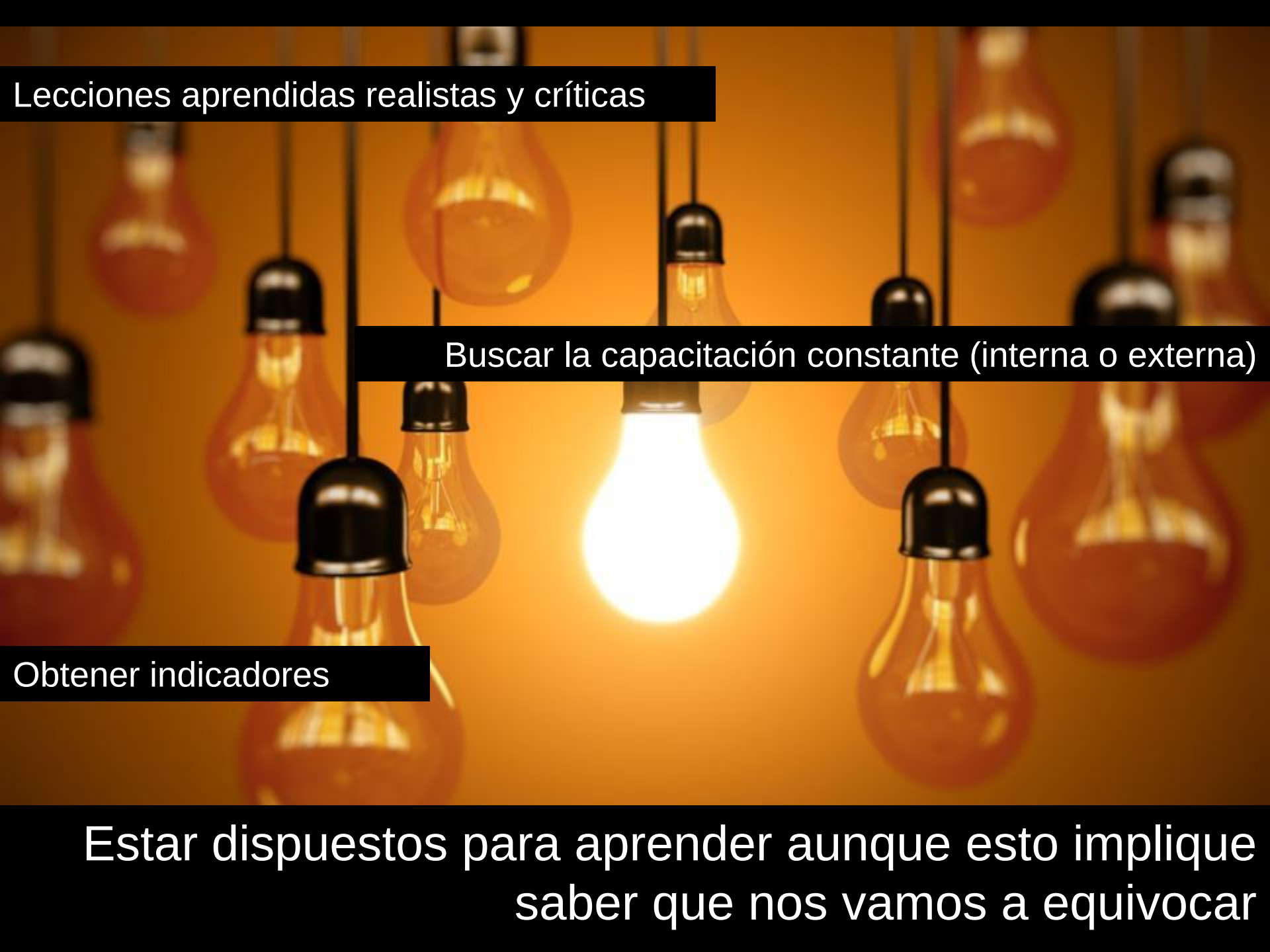
		P	ID	#	Categoría	Severidad	Resolución	Estado	Actualizada	Resumen
☐	✎		0013322	2	[Sección 2. [Localización]] Registro y edición	crítico	Corregida	resuelta (HugoC)	**2015-02-16**	De forma indistinta el sistema presentó " {"json_status":"1"}"
☐	✎		0013509		[Sección 16. [Anexos]] Registro y edición	normal	Corregida	resuelta (HugoC)	**2015-02-16**	Al borrar la información de los datos "Pie de imagen" y dar clic en "Guardar" el sistema la recupera y no la borra
☐	✎		0013883		[Sección 6. [Información histórica]] Registro y edición	bloqueante	Abierta	asignada (karlafm)	2015-01-20	El sistema continua mostrando el dato "Especifique" al desactivar la casilla de verificación
☐	✎		0013884		[Sección 2. [Localización]] Registro y edición	menor	Abierta	asignada (LauraKAH)	2015-01-20	Se sugiere omitir las opciones "Ninguno" de la lista de colonias de la delegación "Tláhuac"
☐	✎		0013895		[General] Página para el registro y edición de una ficha (pestañas)	normal	Abierta	asignada (karlafm)	2015-01-19	En Firefox 34.05 al dar clic en una sección se expande una diferente a la seleccionada
☐	✎		0013896		[3. Flujo de la ficha] Solicitar validación	normal	Abierta	asignada (karlafm)	2015-01-19	En Firefox 34.05 al solicitar la validación de la ficha se presentó error técnico

Mantis

ID	Proyecto	Categoría	Visibilidad	Fecha de Envío	Última actualización
0013322	Sección 2. [Localización]	Registro y edición	privado	2014-12-08 19:12	2015-02-16 13:32

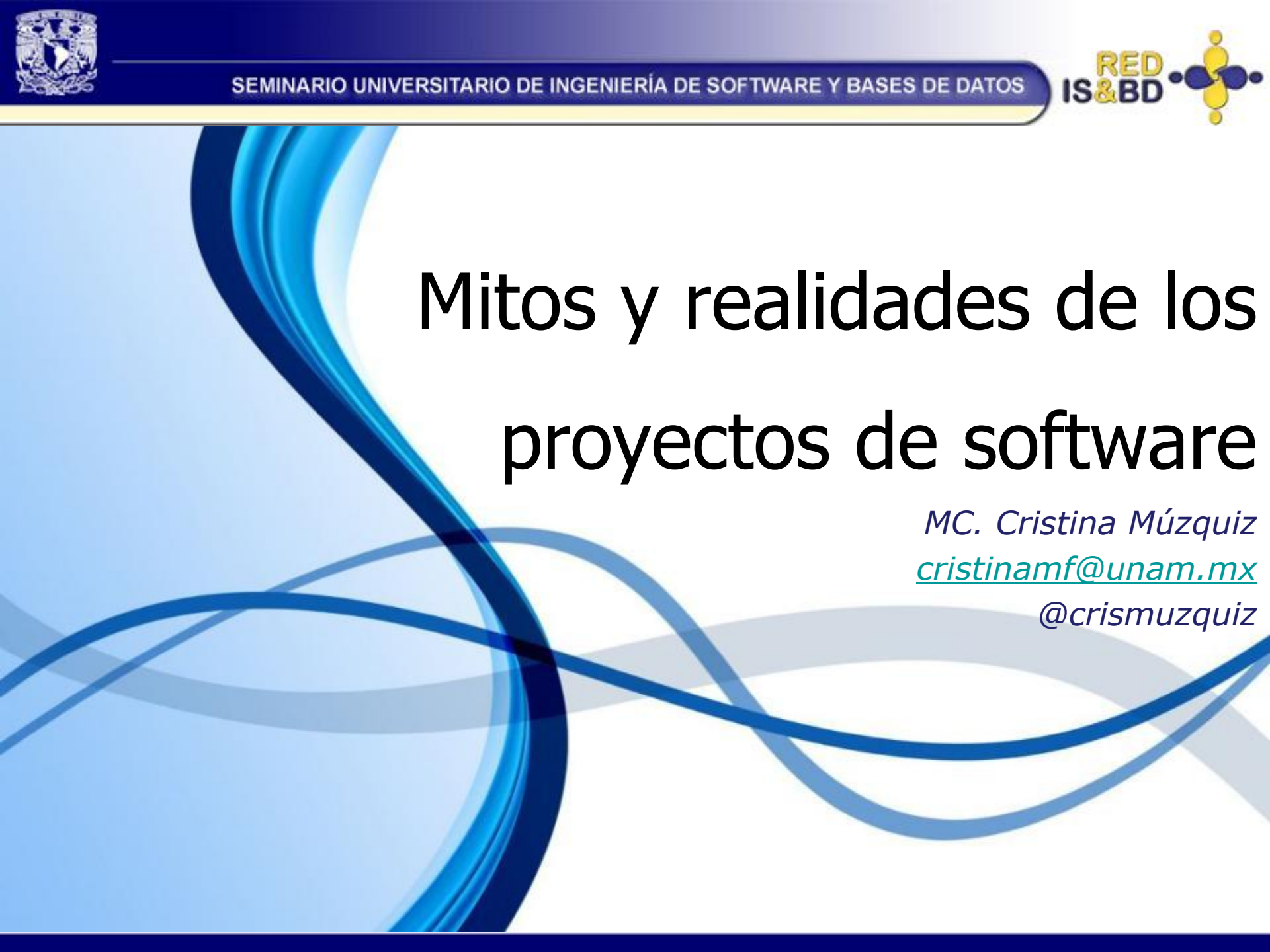
Informador	CristhianAB				
Asignada a	HugoC				
Prioridad	ninguna	**Severidad**	crítico	**Reproducibilidad**	siempre
Estado	resuelta	**Resolución**	Corregida		
Plataforma		**Sistema Operativo**		**Versión de S.O.**	

Resumen	0013322: De forma indistinta el sistema presentó "{"json_status":"1"}"
Descripción	Al realizar movimientos en la ficha, ingresar a la sección, eliminar los datos y dar clic en el botón "Guardar", de forma indistinta se presentó el mensaje "{"json_status":"1"}". El sistema no debería presentar errores.
Pasos para reproducirlo	1. Realizar movimientos a lo largo de la ficha 2. Ir a la sección y eliminar todos los datos del contenido 3. Dar clic en "Guardar"
Información Adicional	http://132.248.63.217/cnmhi_pruebas/RegFicha/guardar/Localizacion [^]



Estar dispuestos para aprender aunque esto implique saber que nos vamos a equivocar

Dudas



Mitos y realidades de los proyectos de software

MC. Cristina Múzquiz
cristinamf@unam.mx
@crismuzquiz