

Arquitectura Orientada a Servicios

Introducción

Edwin Salvador

10 de abril de 2015

Sesión 1

Contenido I

- 1 Aspectos Administrativos
- 2 Sobre mi
- 3 Sobre ustedes
- 4 Sobre la materia
- 5 Políticas del curso
- 6 ¿Cómo serán las prácticas?
- 7 Revisión de las herramientas instaladas
 - Control de versiones

Aspectos Administrativos

Quién?, Cuándo?, Dónde?

- **Profesor:** Ing. Edwin Salvador
- **Email:** edwin.salvador@epn.edu.ec
- **Cuándo?** Viernes de 17 a 20
- **Dónde?** ESFOT 22A
- **Oficina** #3 en el edificio de la dirección de la ESFOT junto al centro de cómputo.

Contenido I

- 1 Aspectos Administrativos
- 2 Sobre mi
- 3 Sobre ustedes
- 4 Sobre la materia
- 5 Políticas del curso
- 6 ¿Cómo serán las prácticas?
- 7 Revisión de las herramientas instaladas
 - Control de versiones

Sobre mi

Edwin Salvador

- Graduado de Ingeniería en Sistemas en la PUCE, 2009.

Sobre mi

Edwin Salvador

- Graduado de Ingeniería en Sistemas en la PUCE, 2009.
- Expositor IV Congreso de Ingeniería de Sistemas, SINAPUCE, 2009.

Sobre mi

Edwin Salvador

- Graduado de Ingeniería en Sistemas en la PUCE, 2009.
- Expositor IV Congreso de Ingeniería de Sistemas, SINAPUCE, 2009.
- Desarrollador web (PHP, MySQL, Javascript, HTML), 2008-2012.

Sobre mi

Edwin Salvador

- Graduado de Ingeniería en Sistemas en la PUCE, 2009.
- Expositor IV Congreso de Ingeniería de Sistemas, SINAPUCE, 2009.
- Desarrollador web (PHP, MySQL, Javascript, HTML), 2008-2012.
- Maestría en Ingeniería de Computación Distribuida en la Universidad de Melbourne, 2012-2014.

- Graduado de Ingeniería en Sistemas en la PUCE, 2009.
- Expositor IV Congreso de Ingeniería de Sistemas, SINAPUCE, 2009.
- Desarrollador web (PHP, MySQL, Javascript, HTML), 2008-2012.
- Maestría en Ingeniería de Computación Distribuida en la Universidad de Melbourne, 2012-2014.
- Desarrollo de sistema para residencias universitarias de UniMelb.

- Graduado de Ingeniería en Sistemas en la PUCE, 2009.
- Expositor IV Congreso de Ingeniería de Sistemas, SINAPUCE, 2009.
- Desarrollador web (PHP, MySQL, Javascript, HTML), 2008-2012.
- Maestría en Ingeniería de Computación Distribuida en la Universidad de Melbourne, 2012-2014.
- Desarrollo de sistema para residencias universitarias de UniMelb.
- Expositor en Symposium on Information Management and Big Data (SIMBIG), 2014, Perú

- Graduado de Ingeniería en Sistemas en la PUCE, 2009.
- Expositor IV Congreso de Ingeniería de Sistemas, SINAPUCE, 2009.
- Desarrollador web (PHP, MySQL, Javascript, HTML), 2008-2012.
- Maestría en Ingeniería de Computación Distribuida en la Universidad de Melbourne, 2012-2014.
- Desarrollo de sistema para residencias universitarias de UniMelb.
- Expositor en Symposium on Information Management and Big Data (SIMBIG), 2014, Perú
- Docente tiempo completo en la ESFOT, 2014 - ?

Interesado en...

- Computación distribuida.

Interesado en...

- Computación distribuida.
- Big Data y análisis de datos

Interesado en...

- Computación distribuida.
- Big Data y análisis de datos
- Minería de datos y minería de procesos

Interesado en...

- Computación distribuida.
- Big Data y análisis de datos
- Minería de datos y minería de procesos
- Datos abiertos

Interesado en...

- Computación distribuida.
- Big Data y análisis de datos
- Minería de datos y minería de procesos
- Datos abiertos
- Programación web HTML5, PHP, node.js, D3.js.

Interesado en...

- Computación distribuida.
- Big Data y análisis de datos
- Minería de datos y minería de procesos
- Datos abiertos
- Programación web HTML5, PHP, node.js, D3.js.
- Python, bases de datos noSQL.

Contenido I

- 1 Aspectos Administrativos
- 2 Sobre mi
- 3 Sobre ustedes**
- 4 Sobre la materia
- 5 Políticas del curso
- 6 ¿Cómo serán las prácticas?
- 7 Revisión de las herramientas instaladas
 - Control de versiones

Sobre ustedes?

- Nombre?
- Confirmar correo (correo EPN Microsoft)
- Trabaja o estudiante tiempo completo?
- Qué es SOA? ¿Experiencia?

Contenido I

- 1 Aspectos Administrativos
- 2 Sobre mi
- 3 Sobre ustedes
- 4 Sobre la materia**
- 5 Políticas del curso
- 6 ¿Cómo serán las prácticas?
- 7 Revisión de las herramientas instaladas
 - Control de versiones

Sobre la materia

- Sílabo

Sobre la materia

- Sílabo
- Bibliografía:

Sobre la materia

- Sílabo
- Bibliografía:
 - Lonneke Dikmans (2012), SOA Made Simple, 1ra edición, Packt Publishing

Sobre la materia

- Sílabo
- Bibliografía:
 - Lonneke Dikmans (2012), SOA Made Simple, 1ra edición, Packt Publishing
 - Nicolai M. Josuttis (2007), SOA in Practice, 1ra edición, O'Reilly

Método de evaluación

- Pruebas parciales **20 %**
- Laboratorios y deberes **30 %**
- Examen bimestral **40 %**
- Participación en clase **10 %**

Presentaciones

- Podemos planificar presentaciones para el segundo bimestre.
- Estas presentaciones pueden ser sobre casos de estudio de alguna empresa que haya implementado SOA.
- ¿Cuál era la situación de la empresa?
- ¿Qué alternativa de solución se implementó?
- ¿Qué tecnología se utilizó?
- ¿En que etapa de implementación se encuentra?
- ¿Qué beneficios obtuvo la empresa?
- ¿Qué cambios tuvo la empresa en cuanto a sus procesos?

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.
- Organización del reporte:

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.
- Organización del reporte:
 - Resumen

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.
- Organización del reporte:
 - Resumen
 - Introducción

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.
- Organización del reporte:
 - Resumen
 - Introducción
 - Desarrollo

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.
- Organización del reporte:
 - Resumen
 - Introducción
 - Desarrollo
 - Conclusiones

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.
- Organización del reporte:
 - Resumen
 - Introducción
 - Desarrollo
 - Conclusiones
 - Recomendaciones

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.
- Organización del reporte:
 - Resumen
 - Introducción
 - Desarrollo
 - Conclusiones
 - Recomendaciones
- Los reportes serán entregados vía Turnitin (creación de cuentas)

Organización de las presentaciones de proyectos

- Todas las presentaciones deben ir acompañadas de un reporte.
- Duración: 15 a 20 min. aprox.
- Organización de la presentación:
 - Agenda
 - Introducción
 - Desarrollo de la presentación
 - Resultados o conclusión.
- Organización del reporte:
 - Resumen
 - Introducción
 - Desarrollo
 - Conclusiones
 - Recomendaciones
- Los reportes serán entregados vía Turnitin (creación de cuentas)
- Fuentes para los reportes o consultas deben ser de la biblioteca, libros físicos o virtuales, artículos científicos verificados, etc (NO Wikipedia, blogs, rincón del vago, monografías, etc)

Contenido I

- 1 Aspectos Administrativos
- 2 Sobre mi
- 3 Sobre ustedes
- 4 Sobre la materia
- 5 Políticas del curso**
- 6 ¿Cómo serán las prácticas?
- 7 Revisión de las herramientas instaladas
 - Control de versiones

Políticas del curso

- No tolerancia a la copia

Políticas del curso

- No tolerancia a la copia
 - Primera vez = 0 en el elemento de evaluación (deber, prueba, etc).

- **No tolerancia a la copia**
 - Primera vez = 0 en el elemento de evaluación (deber, prueba, etc).
 - Segunda vez será llevado a consejo para evaluar la gravedad.

- **No tolerancia a la copia**
 - Primera vez = 0 en el elemento de evaluación (deber, prueba, etc).
 - Segunda vez será llevado a consejo para evaluar la gravedad.
 - Tercera vez reprueba la materia.

Políticas del curso

- **No tolerancia a la copia**
 - Primera vez = 0 en el elemento de evaluación (deber, prueba, etc).
 - Segunda vez será llevado a consejo para evaluar la gravedad.
 - Tercera vez reprueba la materia.
- No se aceptan deberes atrasados.

- **No tolerancia a la copia**
 - Primera vez = 0 en el elemento de evaluación (deber, prueba, etc).
 - Segunda vez será llevado a consejo para evaluar la gravedad.
 - Tercera vez reprueba la materia.
- No se aceptan deberes atrasados.
- Pruebas atrasadas con certificado médico sellado.

- **No tolerancia a la copia**
 - Primera vez = 0 en el elemento de evaluación (deber, prueba, etc).
 - Segunda vez será llevado a consejo para evaluar la gravedad.
 - Tercera vez reprueba la materia.
- No se aceptan deberes atrasados.
- Pruebas atrasadas con certificado médico sellado.
- Se tendrá en cuenta la asistencia a clases.

- **No tolerancia a la copia**
 - Primera vez = 0 en el elemento de evaluación (deber, prueba, etc).
 - Segunda vez será llevado a consejo para evaluar la gravedad.
 - Tercera vez reprueba la materia.
- No se aceptan deberes atrasados.
- Pruebas atrasadas con certificado médico sellado.
- Se tendrá en cuenta la asistencia a clases.
- Las diapositivas estarán disponibles cada semana. Si no las subo deben hacerme acuerdo. No es justificación el no hacer deberes o no estudiar porque las diapositivas no estaban subidas.

Contenido I

- 1 Aspectos Administrativos
- 2 Sobre mi
- 3 Sobre ustedes
- 4 Sobre la materia
- 5 Políticas del curso
- 6 ¿Cómo serán las prácticas?
- 7 Revisión de las herramientas instaladas
 - Control de versiones

¿Cómo serán las prácticas?

- Se procurará hacer ejercicios prácticos todas las clases.

¿Cómo serán las prácticas?

- Se procurará hacer ejercicios prácticos todas las clases.
- Utilizaremos Git para entregar el trabajo realizado en el laboratorio.

¿Cómo serán las prácticas?

- Se procurará hacer ejercicios prácticos todas las clases.
- Utilizaremos Git para entregar el trabajo realizado en el laboratorio.
- Se evaluará únicamente el trabajo realizado dentro del laboratorio.

¿Cómo serán las prácticas?

- Se procurará hacer ejercicios prácticos todas las clases.
- Utilizaremos Git para entregar el trabajo realizado en el laboratorio.
- Se evaluará únicamente el trabajo realizado dentro del laboratorio.
- Si no se alcanza a terminar todos los ejercicios de la práctica, ustedes pueden terminarlo en casa pero no se tomará en cuenta para la evaluación.

¿Cómo serán las prácticas?

- Se procurará hacer ejercicios prácticos todas las clases.
- Utilizaremos Git para entregar el trabajo realizado en el laboratorio.
- Se evaluará únicamente el trabajo realizado dentro del laboratorio.
- Si no se alcanza a terminar todos los ejercicios de la práctica, ustedes pueden terminarlo en casa pero no se tomará en cuenta para la evaluación.
- Al final del semestre revisaré el número de prácticas terminadas, si todas las prácticas están al 100 %, esto significará 1 punto extra a la nota de laboratorios.

Contenido I

- 1 Aspectos Administrativos
- 2 Sobre mi
- 3 Sobre ustedes
- 4 Sobre la materia
- 5 Políticas del curso
- 6 ¿Cómo serán las prácticas?
- 7 Revisión de las herramientas instaladas
 - Control de versiones

Revisión de las herramientas instaladas

- Netbeans

Revisión de las herramientas instaladas

- Netbeans
- Visual Studio

Revisión de las herramientas instaladas

- Netbeans
- Visual Studio
- Java EE 7

Revisión de las herramientas instaladas

- Netbeans
- Visual Studio
- Java EE 7
- GlassFish

Revisión de las herramientas instaladas

- Netbeans
- Visual Studio
- Java EE 7
- GlassFish
- Git

Contenido I

- 1 Aspectos Administrativos
- 2 Sobre mi
- 3 Sobre ustedes
- 4 Sobre la materia
- 5 Políticas del curso
- 6 ¿Cómo serán las prácticas?
- 7 Revisión de las herramientas instaladas
 - Control de versiones

Control de versiones con Git

- Git es un sistema de control de versiones (VCS) de código abierto y distribuido capaz de trabajar con proyectos grandes y pequeños de manera eficiente.

Control de versiones con Git

- Git es un sistema de control de versiones (VCS) de código abierto y distribuido capaz de trabajar con proyectos grandes y pequeños de manera eficiente.
- Git funciona estrictamente mediante la línea de comandos.

Control de versiones con Git

- Git es un sistema de control de versiones (VCS) de código abierto y distribuido capaz de trabajar con proyectos grandes y pequeños de manera eficiente.
- Git funciona estrictamente mediante la línea de comandos.
- Git facilita la interacción con *GitHub* (un repositorio web con una interfaz amigable).

Control de versiones con Git

- Git es un sistema de control de versiones (VCS) de código abierto y distribuido capaz de trabajar con proyectos grandes y pequeños de manera eficiente.
- Git funciona estrictamente mediante la línea de comandos.
- Git facilita la interacción con *GitHub* (un repositorio web con una interfaz amigable).
- Brevemente veamos como instalar y configurar Git en nuestro ambiente de trabajo.

Instalando Git

- Crear una cuenta gratuita en <http://github.com>

Instalando Git

- Crear una cuenta gratuita en <http://github.com>
- Descargar Git bash desde:

Instalando Git

- Crear una cuenta gratuita en <http://github.com>
- Descargar Git bash desde:
 - <http://git-scm.com> (**Trabajaremos con esta**)

Instalando Git

- Crear una cuenta gratuita en <http://github.com>
- Descargar Git bash desde:
 - <http://git-scm.com> (**Trabajaremos con esta**)
- GitHub ofrece una interfaz gráfica (no necesario):

Instalando Git

- Crear una cuenta gratuita en <http://github.com>
- Descargar Git bash desde:
 - <http://git-scm.com> (**Trabajaremos con esta**)
- GitHub ofrece una interfaz gráfica (no necesario):
 - <https://windows.github.com> (Windows)

Instalando Git

- Crear una cuenta gratuita en <http://github.com>
- Descargar Git bash desde:
 - <http://git-scm.com> (**Trabajaremos con esta**)
- GitHub ofrece una interfaz gráfica (no necesario):
 - <https://windows.github.com> (Windows)
 - <https://mac.github.com> (Mac)

Creando un repositorio remoto en GitHub

- Clic en el (+) en la esquina superior derecha (Nuevo repositorio).
- Ingresar nombre del repositorio (esfot_soa)
- Ingresar descripción del repositorio
- Seleccionar Público, no inicializar con README.md, .gitignore = none, license = none.
- Clic en crear

Configurando Git

Empezamos con la configuración del usuario para todos los repositorios locales.

- El nombre que queremos que vaya con nuestras transacciones commit:
`$ git config --global user.name "[nombre]"`

Configurando Git

Empezamos con la configuración del usuario para todos los repositorios locales.

- El nombre que queremos que vaya con nuestras transacciones commit:
`$ git config --global user.name "[nombre]"`
- El email que queremos que vaya con nuestras transacciones commit:
`$ git config --global user.email "[email address]"`

Configurando Git

Empezamos con la configuración del usuario para todos los repositorios locales.

- El nombre que queremos que vaya con nuestras transacciones commit:
`$ git config --global user.name "[nombre]"`
- El email que queremos que vaya con nuestras transacciones commit:
`$ git config --global user.email "[email address]"`
- Habilitar colores en la línea de comandos:
`$ git config --global color.ui auto`

Creando repositorios

Para crear un nuevo repositorio o obtener uno desde una URL ya existente.

- Crear un nuevo repositorio local con un nombre específico:

```
$ git init [project-name]
```

Creando repositorios

Para crear un nuevo repositorio o obtener uno desde una URL ya existente.

- Crear un nuevo repositorio local con un nombre específico:

```
$ git init [project-name]
```

- También podemos inicializar en un directorio ya existente, en este caso ejecutaremos solamente `git init`.

Creando repositorios

Para crear un nuevo repositorio o obtener uno desde una URL ya existente.

- Crear un nuevo repositorio local con un nombre específico:

```
$ git init [project-name]
```

- También podemos inicializar en un directorio ya existente, en este caso ejecutaremos solamente `git init`.

- Descargar un proyecto existente:

```
$ git clone [url]
```


Configurando el repositorio local

- Crear archivo README para el repositorio:

```
touch README.md
```

- Modificar el archivo README creado

```
echo texto descriptivo del repositorio » README.md
```

- Crear .gitignore

```
touch .gitignore
```

- Modificar y añadir los archivos que se desean excluir del versionamiento. (*.pdf, *.tex, etc)

Realizar cambios

Revisar las ediciones y empezar una transacción `commit`.

- Listar todos los archivos nuevos o modificados que necesitan ser subidos:

```
$ git status
```

Realizar cambios

Revisar las ediciones y empezar una transacción `commit`.

- Listar todos los archivos nuevos o modificados que necesitan ser subidos:

```
$ git status
```

- Mostrar las modificaciones en los archivos que no han sido subidas:

```
$ git diff
```

Realizar cambios

Revisar las ediciones y empezar una transacción `commit`.

- Listar todos los archivos nuevos o modificados que necesitan ser subidos:

```
$ git status
```

- Mostrar las modificaciones en los archivos que no han sido subidas:

```
$ git diff
```

- Añadir un archivo al control de versiones:

```
$ git add [archivo]
```

Realizar cambios

Revisar las ediciones y empezar una transacción `commit`.

- Listar todos los archivos nuevos o modificados que necesitan ser subidos:

```
$ git status
```

- Mostrar las modificaciones en los archivos que no han sido subidas:

```
$ git diff
```

- Añadir un archivo al control de versiones:

```
$ git add [archivo]
```

- Mostrar las diferencias entre el archivo a subir y la última versión subida:

```
$ git diff --staged
```

Realizar cambios

Revisar las ediciones y empezar una transacción `commit`.

- Listar todos los archivos nuevos o modificados que necesitan ser subidos:

```
$ git status
```

- Mostrar las modificaciones en los archivos que no han sido subidas:

```
$ git diff
```

- Añadir un archivo al control de versiones:

```
$ git add [archivo]
```

- Mostrar las diferencias entre el archivo a subir y la última versión subida:

```
$ git diff --staged
```

- Quitar el archivo del control de versiones pero conservar su contenido:

```
$ git reset [archivo]
```

Realizar cambios

Revisar las ediciones y empezar una transacción `commit`.

- Listar todos los archivos nuevos o modificados que necesitan ser subidos:

```
$ git status
```

- Mostrar las modificaciones en los archivos que no han sido subidas:

```
$ git diff
```

- Añadir un archivo al control de versiones:

```
$ git add [archivo]
```

- Mostrar las diferencias entre el archivo a subir y la última versión subida:

```
$ git diff --staged
```

- Quitar el archivo del control de versiones pero conservar su contenido:

```
$ git reset [archivo]
```

- Subir el archivo y los cambios al historial de versiones permanentemente:

```
$ git commit -m "[mensaje descriptivo]"
```

Subir los cambios

- Vincular con el repositorio remoto (en GitHub)

```
git remote add origin https://github.com/NOMBRE_DE_USUARIO/NOMBRE_DEL_REPOSITORIO.git
```

- Subir los cambios al repositorio remoto

```
git push -u origin master
```

- INGRESAR USUARIO
- INGRESAR CONTRASEÑA
- Para evitar que nos pida usuario y contraseña cada vez, seguimos estos pasos:
<https://help.github.com/articles/generating-ssh-keys/>

Modificando nombres de archivos

Mover o eliminar archivos con control de versiones.

- Eliminar un archivo del directorio actual y registrar la eliminación.

```
$ git rm [archivo]
```

Modificando nombres de archivos

Mover o eliminar archivos con control de versiones.

- Eliminar un archivo del directorio actual y registrar la eliminación.
\$ git rm [archivo]
- Eliminar el archivo del control de versiones pero mantener la copia local.:
\$ git rm --cached [archivo]

Modificando nombres de archivos

Mover o eliminar archivos con control de versiones.

- Eliminar un archivo del directorio actual y registrar la eliminación.

```
$ git rm [archivo]
```

- Eliminar el archivo del control de versiones pero mantener la copia local.:

```
$ git rm --cached [archivo]
```

- Cambiar el nombre del archivo o moverlo a otro directorio:

```
$ git mv [archivo-original] [archivo-renombrado]
```