

# Laboratorio de Computación IV

## Clase 1

*Andrés Fortier*

# Antes de comenzar



- Último semestre de la tecnicatura.
- Tienen una serie de conceptos básicos, los vamos a poner en práctica en un proyecto.
- Su primera responsabilidad:
  - Linux.
  - Conexión a internet.

# Antes de comenzar



- Las transparencias no van a ser tan detalladas como el año pasado
  - Tomar notas.
  - Internet + traducción si lo necesitan.
- Trabajen desde el día uno. Asumo que van a estar al día con lo pedido.
- Típeo en clase.
- Repasar las transparencias al otro día.

# Antes de comenzar

---

- Página:
  - <http://andres-fortier.github.io/laboratorio4-2016/>
- Foro (no escriban al mail personal):
  - [utn-sma-poo@googlegroups.com](mailto:utn-sma-poo@googlegroups.com)

# Objetivos de la materia



- Desarrollar una aplicación web que contemple un conjunto de funcionalidades típicas.
- Exponerlos a un conjunto de herramientas comunes dentro de la industria del desarrollo de software.
- Poner un sistema en producción en un servidor de acceso público y mantenerlo.
- Conceptos generales de aplicaciones web que les pueden preguntar en una entrevista.

# Contenidos de la materia



- Aplicaciones web
  - Protocolo HTTP.
  - Un poco de HTML, CSS y Javascript.
  - Servidores web.
  - MVC en la web.
  - Autenticación y conceptos básicos de seguridad.
  - Autorización. Roles de usuarios y administradores.
  - Persistencia y evolución incremental.
  - Testing de aplicaciones web.

# Contenidos de la materia



- Herramientas:
  - Versionamiento de código: Git y Github.
  - (Opcional) Uso de tickets para la organización del trabajo. Desarrollo de software incremental.
  - Interfaz de línea de comandos: Terminales linux, SSH y comandos comunes.
  - Editores de texto de línea de comando e IDEs.


# Contenidos de la materia

---

- Utilización de un servicio de hosting para colocar la aplicación en producción (Openshift o Heroku).




# Aprobación y promoción



- Centramos la materia en un trabajo que engloba los contenidos descriptos
  - No hay parciales o finales.
  - Tres entregas.
  - Se aprueba y promociona con promedio mayor o igual a 5.
  - Sólo se permite una entrega con nota menor a 5.
- Primera entrega individual.
- Grupos de 2 personas para la aplicación web.

# Aprobación y promoción




- Fechas de entrega:
  - 19/04/2016.
  - 31/05/2016.
  - 02/07/2016.
- 05/07/2016 - Reunion de 30' con cada grupo para charlar sobre su entrega.
- 07/07/2016 - Cierre del cuatrimestre.

# Sobre las entregas



- Código por medio de Github. Se toma lo publicado en “master” antes de que comience el día de la entrega
  - Ej: para 02/07/2016 se toma hasta 01/07/2016 23:59:59.
- 2da y 3ra entrega: sumar demo publicado en Openshift o Heroku.
- Utilicen el sentido común para las entregas.

# Organización de las clases



- Comando linux semanales.
- Conceptos relacionados a desarrollo de aplicaciones web y/o Ruby.
- Opcional: alguna herramienta.
- “Tarea” para la clase que viene en base a lo visto.

# Sobre las aplicaciones web



- Se desarrollan en grupos de 2.
  - Distribución de tareas vs. conocimiento del sistema.
- Deben implementar la funcionalidad mínima y una opcional (nota 5).
- Funcionalidades extras consensuadas suman puntos.
- Distintas complejidades.
- Carta de presentación.
- Pueden armar un modelo de negocios.

# Denuncia de baches/grafitis/etc.


- Estilo <http://www.buenosairesbache.com/>.
- Idealmente aplicación web móvil y desktop.
- Técnicamente compleja
  - Emulación de la cámara y el GPS (html 5).
  - Upload de archivos.
  - Mapas
- Impacto en la sociedad.

# Denuncia de baches/grafitis/etc.

---

- Opcionales
  - Funcionamiento offline.
  - Reportes: queries y exportación a CSV.
  - Votos por usuarios.
  - “Likes” y publicación en redes sociales.


# Registro de niñeras/personal



- Trabajos sensibles donde la recomendación y el boca a boca pesan.
- Perfiles de las personas que brindan servicios.
- Costos por hora.
- Buscador básico.
- Recomendaciones y “reviews”.



# Registro de niñeras/personal



- Opcionales
  - Manejo de horarios (disponibles y necesitados).
  - Buscador avanzado.
  - Notificación (ej. buscar al final del día).
  - Cálculos de costo por hora en base a estadísticas.

# Facturación de prestaciones médicas

---

- Los médicos atienden en forma particular distintos tipos de consultas:
  - Obra social. Cada OS paga un monto por consulta y se factura al principio del mes siguiente. Se pagan a un determinado plazo (30, 60 o 90 días). Todos los meses cambian los valores.
  - Particular. Se paga en el momento.

# Facturación de prestaciones médicas

---

- ABM de consultas.
- Hacer el cierre del mes.
- Ver pagos pendientes y registrar pagos.
- Búsquedas (ej. por paciente).

# Facturación de prestaciones médicas

---

- Opcionales
  - Impresión para la presentación de la facturación en el colegio médico.
  - Múltiples formas de liquidar una obra social.
  - Análisis histórico de rendimientos con alguna librería de gráficos.
  - Exportar datos a CSV y/o Excel.

# Equipos de desarrollo distribuidos



- Sincronizar a programadores distribuidos.
- Concepto de standup (que hice ayer / que voy a hacer hoy / blockers).
- Trazabilidad y estadísticas.
- Dashboard.
- Complejidad en trabajar en un dominio que no manejan.

# Equipos de desarrollo distribuidos



- Opcionales
  - Enviar por mail.
  - Integrar herramientas externas (ej. github).
  - Manejo de grupos.
  - “Pausar” un item.

# Todas las aplicaciones - mínimo



- Desarrollaremos sobre Ruby, usando RoR (Ruby on Rails).
- Funcionalidad de usuario básica
  - Usuarios y cuentas (id + password).
  - Backend de administración.
  - Al menos dos niveles de visibilidad, según la app
    - Roles de usuarios.
    - Usuario anónimo vs. Registrado.


# Todas las aplicaciones - mínimo



- A nivel implementación
  - Utilización de un estilo (CSS) en forma consistente.
  - Tests del modelo.
  - Persistencia
    - Mantener información.
    - Migrar información conforme la app crece.



# Todas las aplicaciones - opcional



- Algunos items dependen de la aplicación.
- Social login (facebook, twiter, gmail, etc).
  - Múltiples perfiles por usuario
- Google maps.
- File uploads.
- Javascript.
- Charts.

# Todas las aplicaciones - opcional

---

- Notificaciones por e-mail.
- Log de errores.
- Tests de UI.
- Tests de integración.
- Métricas de código.
- Análisis de seguridad.
- Cosas que los motiven, bienvenidas sean.

# ¿Preguntas?

---



# Comando(s) de la semana: cd

- cd == change directory.
- Se utiliza para cambiar el **directorio actual** de trabajo.

```
[andres@CrazyGoat:~] █
```

# Comando(s) de la semana: cd



- Ejemplos:
  - `cd .`
  - `cd ..`
  - `cd /home`
  - `cd ~` (o simplemente ``cd``)
  - `cd ../../home`
  - `cd -`
  - `cd /ho<tab>`

# Comando(s) de la semana: ls

- ls == list directory contents.
- Ejemplos
  - `ls` → Nombre de archivos y directorios.
  - `ls -l` → Información sobre archivos y directorios.
  - `ls -lh` → Anterior pero con formateo de tamaño.
  - `ls -a` → Incluir archivos ocultos.
  - `ls -r` → Lista en orden reverso.
  - `ls -lS` → Ordena alfabéticamente.
  - `ls -d` → Sólo subdirectorios.

# Comando(s) de la semana: ls

---

- Los modificadores se pueden combinar:
  - ``ls -lSh``
  - ``ls -lhar``
- Para ver todas las opciones ``man ls``.

# Tarea para el hogar

---

- Instalar linux (si no lo quieren hacer nativo lo pueden hacer en una máquina virtual).
- Tener a mano un editor de texto (ej. gedit).