TALLER DE TECNOLOGÍAS DE PRODUCCIÓN DE SOFTWARE
(Opción D)

Año 2022

Carrera/ Plan:

Analista Programador Universitario P Plan 2021/Plan 2015/Plan 2012

Año: 3
Régimen de Cursada: *Semestral*
Carácter: optativa
Correlativas:
Diseño de Bases de Datos, Algoritmos y Estructuras de Datos, Introducción a los Sistemas Operativos, Orientación a Objetos 1, Taller de lecto comprensión y traducción en Inglés, Ingeniería de Software 2

Profesor/es: Christian Rodriguez
Hs. semanales teoría: 3
Hs. semanales práctica: 3

FUNDAMENTACIÓN

Dentro del marco de la carrera, esta opción propone ampliar los conocimientos con los que cuentan los alumnos abordando las metodologías ágiles para la producción de software. En su contenido, se incluyen planificación de requerimientos, técnicas de testing y programación basada en el lenguaje Ruby. Todas estas componentes se combinan en un entorno muy apropiado donde los alumnos podrán experimentar cada una de las etapas reales del desarrollo que van desde el análisis, desarrollo, prueba y puesta en producción.

OBJETIVOS GENERALES

Introducir a los alumnos en un esquema de organización de producción de software, utilizando metodologías, prácticas y herramientas actualizadas y acordes con los estándares actuales.

Fomentar la práctica del alumno en esquemas de trabajo similares a los que se utilizan en las empresas de desarrollo de productos de software.

Ofrecer a los alumnos alternativas tecnológicas, siempre en base a herramientas de utilización actual en el mercado laboral.

RESULTADOS DE APRENDIZAJE

Aplicar los objetivos generales en proyectos desarrollados con el lenguaje Ruby. Conocer las herramientas necesarias para profundizar y extender sus conocimientos de cara al desempeño profesional en un equipo de desarrollo productivo.

CONTENIDOS MINIMOS

- Introducir un ambiente de desarrollo de software estandarizado (con herramientas integradas que den una visión homogénea y estandarizada de las aplicaciones, su interfaz grafica, el acceso a las bases de datos y la interconexión entre aplicaciones), enfocado a un organismo o “clase” de empresa usuaria.
- Practicar como usar el ambiente de desarrollo y las diferencias que tiene que con el ambiente de producción, ilustrando la metodología organizacional del pasaje de desarrollo a producción.

- Practicar con documentación estandarizada (por ej. historias de usuario) mostrando como se pasa de una especificación al código ejecutable.
- Ejemplificar la actividad del tester de aplicaciones. Metodología de trabajo y ambiente de prueba (diferencia con los otros ambientes)
- Plantear el proceso estandarizado de desarrollo de software en una tecnología de uso en el mercado. Rol de la documentación en cada etapa.
- Plantear el desarrollo de una solución a un problema real y que ilustre todas las problemáticas antes descriptas
- Describir cuales son las principales características de un proceso de desarrollo de software con calidad.

PROGRAMA ANALÍTICO

1. Introducción a las metodologías ágiles
2. Sistemas de control de versiones. Uso de Git
3. Introduciendo Ruby
4. Máquinas virtuales de ruby: MRI, Rubinius, jRuby, etc. RVM
5. El lengaje Ruby: objetos, atributos, clases, variables, colecciones, bloques e iteradores
6. Herencia, módulos y mixins
7. Tipos estándar, métodos, expresiones, excepciones
8. Rake
9. Unit Testing
10. Gems, rubygems, bundler
11. Ruby on rails

BIBLIOGRAFÍA

- H. Kniberg, Scrum and XP from the Trenches, InfoQ
- K. Schwaber/M. Beedle , Agile Software Development with Scrum
- E. Gamma/R. Helm/R. Johnson/J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley
- D. Thomas, Programming Ruby 1.9, The Pragmatic Programmers' Guide
- G. Brown, Ruby Best Practices, O'Reilly
- S.Ruby/D. Thomas/D. Hansson, Agile Web Development with Rails, The Pragmatic Programmers' Guide
- A. Harris/K. Haase, Sinatra: up and running, O'Reilly

BIBLIOGRAFÍA COMPLEMENTARIA

- Rails Guides: http://guides.rubyonrails.org/
- Ruby doc: http://www.ruby-doc.org/

METODOLOGÍA DE ENSEÑANZA

La metodología es del tipo taller, con clases teóricas donde se desarrollan los aspectos conceptuales del lenguaje, que se dictan utilizando presentaciones del tipo slideshow online y muchos ejemplos prácticos compartiendo la pantalla de la terminal del docente.

La presentación utilizada en las teorías están disponibles online desde el comienzo de las clases y las mismas contienen ejemplos de código que pueden descargarse. Los ejemplos son completos, y funcionales, permitiendo al alumno resolver distintas situaciones problemáticas ilustrando conceptos complejos.

Durante el dictado de las clases teóricas, se utiliza el proyector en el aula, se ejemplifica en la PC del docente y utiliza la pizarra para diagramar o explicar casos concretos. Las clases prácticas por su parte, se realizan en la sala de PC. Los estudiantes plantean sus dudas y trabajan con los ayudantes, quienes los acompañan en este proceso. El JTP es quien realiza explicaciones de práctica al inicio de cada trabajo práctico, haciendo hincapié en los ejercicios más importantes y puntos a evaluar. En total se deben completar 6 prácticas.

Algunas de estas prácticas contienen entregas que los alumnos deberán entregar. Éstas entregas consideran una instancia de coloquio donde el docente a cargo del grupo realiza distintas preguntas sobre la temática abordada. La entrega se realiza a través de la plataforma virtual.

Dentro de la plataforma Moodle se incluyen autoevaluaciones de carácter optativo para reforzar temas y obligatorios. También se plantean entregas con fecha límite donde los alumnos pueden subir la versión finalizada de sus trabajos para la posterior corrección del pantel docente.

Cada ayudante tiene a cargo un grupo de alumnos, y será su función seguirlo en el desenvolvimiento de la cursada, intentado identificar los puntos problemáticos para poder resolverlos en forma rápida y no provoque el abandono de la cursada.

Las entregas que realizan los alumnos se complementan con coloquios entre el estudiante que expone la tarea realizada en forma individual y el docente. Es en esta instancia donde se evalúa no sólo el desarrollo, sino además los conocimientos, claridad de su entrega, organización y expresión. Esto se refleja en planillas que conforman documentación de evaluación del coloquio.

A fin de mejorar la comunicación entre los estudiantes y la cátedra, se utilizarán la plataforma con foros de notificaciones unidireccionales de la cátedra hacia los alumnos y por canales de consultas bidireccionales donde los alumnos realizan consultas y la cátedra responde. De esta forma la comunicación fluye de forma ágil y automática.

EVALUACIÓN

La **aprobación de la cursada** estará dada por la aprobación del 80% de los trabajos prácticos y el trabajo final integrador.

En la cátedra se pone énfasis en la capacidad del alumno para conocer técnicas y herramientas de aplicación en Informática, en lo posible siguiendo las tendencias marcadas por el software libre, y en la aplicación efectiva de las mismas. La cátedra acompaña el proceso con materiales para que el alumno estudie casos, valore la selección y empleo eficiente de herramientas, y técnicas determinadas para cada problema. La evaluación de esta competencia forma parte de las evaluaciones de trabajos prácticos.

La **aprobación de la materia** estará dada por la aprobación de la cursada y un extensión del trabajo final de cursada. La asistencia a las clases teóricas aportará a la calificación final.

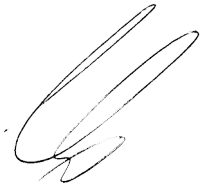
CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	21/8	Introducción Metodologías ágiles Uso de GIT
2	28/8	Introducción a Ruby
3	4/9	VMS Ruby Instalación del entorno
4	11/9	Objetos y atributos Clases Variables
5	18/9	Colecciones
6	25/9	Bloques e iteradores
7	2/10	Herencia, módulos y mixins
8	9/10	Tipos estándar Métodos Expresiones
9	16/10	Excepciones
10	23/10	Gemas Bundler
11	30/10	ORMs Active Record
12	6/11	Framework Rails
13	13/11	Framework Rails

Evaluaciones previstas	Fecha
Objetos y atributos. Clases. Variables. Colecciones. Bloques e iteradores. Herencia, módulos y mixins. Tipos estándar. Métodos. Expresiones	23/10
Excepciones. Gemas. ORMs	13/11
Trabajo práctico integrador usando Rails	5/2

Contacto de la cátedra:

- **Mail:** car@info.unlp.edu.ar
- **Sitio Web:** http://ttps-ruby.github.io/
- **Plataforma virtual:** http://catedras.info.unlp.edu.ar
- **Otros:** https://github.com/ttps-ruby



Lic. Christian Rodriguez