Fundamentos del Software: introducción a la programación verificada

Introducción al curso

Roberto Blanco[†] & Ricardo J. Rodríguez[‡]

MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

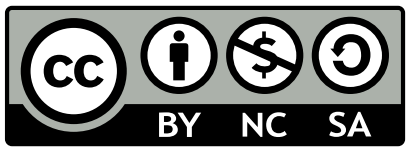
Universidad Zaragoza

[†]Max Plank Institute for Security and Privacy
Bochum, Alemania

[‡]Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza (España)

Septiembre 2020

Universidad de Zaragoza



Índice

Universidad Zaragoza

Índice

Universidad Zaragoza

Profesorado

Roberto Blanco

- Investigador en MPI Security and Privacy (Bochum, Alemania)
- Profesor encargado de programación verificada
- Líneas de investigación:
  - Compilación segura
  - Lógica computacional y lenguajes de programación
  - Verificación y seguridad de sistemas informáticos
- Más información:
  - `https://secure-compilation.github.io/`
- Contacto: `roberto.blanco@csp.mpg.de`

Ricardo J. Rodríguez

- Profesor en Universidad de Zaragoza (España)
- Profesor encargado de programación segura
- Líneas de investigación:
  - Análisis de programas binarios
  - Análisis forense digital
  - Seguridad de sistemas
- Más información:
  - `https://www.reversea.me`
- Contacto: `rjrodriguez@unizar.es`

Universidad Zaragoza 1542

Índice

Universidad Zaragoza

Temario

- **30 horas de curso** + 20 trabajos y estudio
- **Clases de 09:00 a 17:30, de 90 minutos**
- Las mañanas dedicadas a programación verificada y las tardes a programación segura

	Lunes	**Martes**	**Miércoles**	**Jueves**	**Viernes**
09:00-10:30	PV (sesión 1)	PV (sesión 4)	PV (sesión 7)	PV (sesión 10)	PV (sesión 13)
11:00-12:30	PV (sesión 2)	PV (sesión 5)	PV (sesión 8)	PV (sesión 11)	PV (sesión 14)
14:00-15:30	PV (sesión 3)	PV (sesión 6)	PV (sesión 9)	PV (sesión 12)	PV (sesión 15)
16:00-17:30	PS (sesión 1)	PS (sesión 2)	PS (sesión 3)	PS (sesión 4)	PS (sesión 5)

Leyenda: PV=Programación Verificada ; PS= Programación Segura

Universidad Zaragoza

Temario detallado

Programación verificada

1. Programas funcionales. Inducción estructural. Estructuras de datos: listas
2. Programación genérica: polimorfismo. Tácticas de demostración
3. Lógica computacional
4. Proposiciones inductivas (I)
5. Proposiciones inductivas (II)
6. Diccionarios. Semántica de los lenguajes imperativos (I)
7. Semántica de los lenguajes imperativos (II)
8. Automatización de demostraciones
9. Caso de estudio. Proyectos de curso
10. Verificación funcional. Ordenación

Universidad Zaragoza

Temario detallado

Programación verificada (continuación)

11. Árboles binarios. Encapsulación. Generación de código verificado
12. Árboles rojinegros. Automatización práctica
13. Verificación imperativa. Lógica de Hoare (I)
14. Lógica de Hoare (II). Verificación del lenguaje C
15. Estado del arte del software verificado

Programación segura

1. Motivación y estándares
2. Lenguajes de programación. Lenguaje C
3. Cadenas. Enteros
4. Salidas con formato. Condiciones de carrera
5. Laboratorio

Universidad Zaragoza

Índice

Universidad Zaragoza

Evaluación

- **Certificado de asistencia**
  - **Necesario** acudir al 85 % de las sesiones
  - Se proporcionarán los vídeos de las sesiones grabadas a aquellos alumnos que lo soliciten
- **Créditos de Libre Elección**
  - Curso reconocido con **1.5 créditos ECTS** como “Actividad académica complementaria de Grado” y como “Actividad transversal de Doctorado” por la Universidad de Zaragoza
  - **Necesario entregar el trabajo realizado durante el curso**
    - Dos trabajos: sesión 9 PV y sesión 5 PS
- **Encuesta de satisfacción anónima** al finalizar el curso
  - Necesaria para conocer el desarrollo del curso

Universidad Zaragoza

Índice

Universidad Zaragoza

Logística del curso

- **Plataforma Zulip** para comunicación asíncrona
  - `https://funsoft2020.zulipchat.com`
- **Clases por videoconferencia con Google Meet**
  - `https://meet.google.com/kpz-ketq-kfx`
- **Repositorio de materiales**
  - `https://github.com/robblanco/unizar2020`
- **Materiales adicionales del curso**:
  - Coq: `https://coq.inria.fr/download`, `https://x80.org/rhino-coq/`
  - Máquina virtual para programación segura: `https://drive.google.com/drive/folders/1QCZrDzoptIbkisWJ52Y6_z_KZEh1i4V0?usp=sharing`
    - Como software hipervisor, cualquiera (e.g. VirtualBox)

Universidad Zaragoza
1542

Índice

Universidad Zaragoza

Bibliografía

Programación verificada

- *Software foundations*, varios autores, publicación online en `https://softwarefoundations.cis.upenn.edu/`, 2020
- *Interactive theorem proving and program development*, Yves Bertot y Pierre Castéran, Springer, 2004
- *Certified programming with dependent types*, Adam Chlipala, MIT Press, 2013

Programación segura

- *Effective C: An Introduction to Professional C Programming*, Robert C. Seacord, No Scratch, 9781718501041, Agosto 2020
- *Secure Coding in C and C++*, Software Engineering Institute, Carnegie Mellon University, 2014
- *Evaluation of CERT Secure Coding Rules through Integration with Source Code Analysis Tools*, S. Dewhurst et al., Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2008-TR-014, 2008

Fundamentos del Software: introducción a la programación verificada

Introducción al curso

Roberto Blanco[†] & Ricardo J. Rodríguez[‡]

MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

Universidad Zaragoza

[†]Max Plank Institute for Security and Privacy
Bochum, Alemania

[‡]Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza (España)

Septiembre 2020

Universidad de Zaragoza