

---

**Formulario de aprobación de curso de posgrado/educación permanente**

**Asignatura:**

Centro de Ensayos de Software: Calidad de Código y Testing

**Modalidad:**

(posgrado, educación permanente o ambas)

**Posgrado**

**Educación permanente**

---

**Profesor de la asignatura <sup>1</sup>: Ph.D, Beatriz Pérez Lamancha, Responsable de Calidad, Indaba Consultores S.L. - España**

(título, nombre, grado o cargo, instituto o institución)

**Profesor Responsable Local <sup>1</sup>: Dra. Ing. Laura González, Grado 4, Instituto de Computación**

(título, nombre, grado, instituto)

**Otros docentes de la Facultad:**

(título, nombre, grado, instituto)

**Docentes fuera de Facultad:**

(título, nombre, grado, instituto)

<sup>1</sup> Agregar CV si el curso se dicta por primera vez.

(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

**Instituto o unidad:** Centro de Ensayos de Software

**Departamento o área:**

---

**Horas Presenciales: 12 hs (a distancia, sincrónicas)**

(se deberán discriminar las horas en el ítem Metodología de enseñanza)

**Nº de Créditos: no corresponde**

[Exclusivamente para curso de posgrado]

(de acuerdo a la definición de la UdelaR, un crédito equivale a 15 horas de dedicación del estudiante según se detalla en el ítem Metodología de enseñanza)

**Público objetivo:**

- Programadores que quieren mejorar su forma de desarrollar software.
- Jefes de desarrollo que quieren conocer cómo está desarrollado el software, qué cobertura de pruebas se tiene, qué complejidad o nivel de duplicidad y el costo de mejorarlo.
- Proveedores de servicios de desarrollo de software interesados en mejorar la calidad del servicio que entregan.
- Testers interesados en la cobertura de las pruebas y de la aplicación de buenas prácticas que mejoren la calidad del software.
- Consumidores de software desarrollado por terceros que quieren poder comparar el software desarrollado por distintos proveedores.

**Cupos:** sin cupo

(si corresponde, se indicará el número de plazas, mínimo y máximo y los criterios de selección. Asimismo, se adjuntará en nota aparte los fundamentos de los cupos propuestos. Si no existe indicación particular para el cupo máximo, el criterio general será el orden de inscripción, hasta completar el cupo asignado)

---

### Objetivos:

El objetivo de este curso es aprender a construir código de calidad y limpio a través de la herramienta de código abierto SonarQube.

Automatizando con SonarQube se obtiene una visión global de la calidad del software, junto con pautas claras de cómo mejorar la confiabilidad y seguridad de la aplicación.

Se dan a conocer buenas prácticas de código junto con reglas y métricas que SonarQube aporta para su instauración.

---

### Conocimientos previos exigidos:

Tener conocimientos en programación.

Otros requisitos:

- Disponer de 15 a 20 horas semanales para participar en la formación.
- Tener acceso a un PC con Internet y disponer de parlantes y micrófono.

---

### Conocimientos previos recomendados:

No aplica

---

### Metodología de enseñanza:

El curso se dicta en modalidad en línea.

Descripción de la metodología:

La metodología de enseñanza conjuga elementos de aprendizaje tradicional y de aprendizaje basado en problemas (ABP). El aprendizaje es un proceso constructivo y no receptivo por lo que se plantean problemas que llevan al alumno a comprender mejor el marco teórico para tratar de resolverlos.

El material teórico, que consiste de lecciones previamente preparadas y bibliografía adicional, se pone a disposición del estudiante para su estudio individual. Posteriormente se dictan clases sobre cada tema con el objetivo de aclarar las dudas y/o profundizar sobre aspectos particulares que se considere pertinentes.

El curso tiene un fuerte componente práctico. En cada tema se presentan actividades que permiten a los estudiantes plantear interrogantes, investigar y trabajar en equipo para aplicar el conocimiento teórico y aprender. Estas actividades pueden ser individuales o grupales. Las actividades son corregidas y evaluadas por el docente que devuelve al alumno los comentarios correspondientes para mejorar su rendimiento.

El objetivo de la evaluación es verificar que los estudiantes asimilaron y son capaces de aplicar lo aprendido a problemas reales y no que salven la Prueba final repitiendo información aprendida de memoria.

Detalle de horas:

- Horas de clase (teórico): 5 (videoconferencia, sincrónicas)
- Horas de clase (práctico): 5 (videoconferencia, sincrónicas)
- Horas de clase (laboratorio): 0
- Horas de consulta: 0
- Horas de evaluación: 2
  - Subtotal de horas presenciales: 12 horas sincrónicas (videoconferencia).
- Horas de estudio: 8 horas
- Horas de resolución de ejercicios/prácticos: 40 horas (incluyen horas de consultas en foros)
- Horas proyecto final/monografía: 0

- Total de horas de dedicación del estudiante: 60 horas
- 

### Forma de evaluación:

La evaluación se realizará mediante actividades obligatorias y una prueba final.

### Aprobación

Para aprobar el curso se deberá:

- Entregar y participar de toda actividad obligatoria y obtener en promedio un 60% de los puntos de las actividades obligatorias,
- Obtener al menos 60% de los puntos de la prueba final

La nota de cada curso se calcula con el siguiente criterio:

- 50% nota de prueba final,
  - 40% nota promedio de todas las actividades del curso,
  - 10% rendimiento individual evaluado por el docente, que incluye responsabilidad, motivación, interés, prolijidad, participación de actividades opcionales, foros, videoconferencias (participación sincrónica o asincrónica).
- 

### Temario:

- Introducción
  - Análisis estático de código
  - Metodología de trabajo con SonarQube
  - Buenas prácticas de código y sus reglas de SonarQube
  - Administración de los proyectos en SonarQube
- 

### Bibliografía:

2003, Kent B. "Test-Driven Development by example"  
2009, Rober C. Martin, "Clean Code"  
2010, Pressman S. Roger. "Ingeniería del software. Un enfoque práctico"  
2011, Myers G., Sandler C., Badgett T. "The Art of Software Testing"  
2011, Sommerville I. "Ingeniería de Software"  
2013, Wiegers K., Beatty J. "Software Requirements. Best practices"  
2014, Spillner, A., Linz, T., Schaefer, H. "Software Testing Foundations"  
2014, SWEBOK. "Guide to the Software Engineering Body of Knowledge"

---

**Datos del curso**

---

**Fecha de inicio y finalización: setiembre 2023**

**Horario y Salón:**

No aplica

**Arancel: \$ 22.000 (pesos uruguayos veintidós mil)**

[Si la modalidad no corresponde indique "no corresponde". Si el curso contempla otorgar becas, indíquelo]

**Arancel para estudiantes inscriptos en la modalidad posgrado: no corresponde**

**Arancel para estudiantes inscriptos en la modalidad educación permanente: \$ 22.000 (pesos uruguayos veintidós mil)**

---