



Javier Sevilla Ballesteros

Graduado en Ingeniería Informática

contacto

javiersevball@gmail.com
javiersevball.github.io/
javiersevball

Ciudad Real, España

idiomas

Inglés [B1 / UCLM]
Castellano [nativo]

estudios universitarios

Sep 2012
Feb 2017

Grado en Ingeniería Informática
Euro-Inf Quality Label

Universidad de Castilla-La Mancha

Intensificación de **Ingeniería de Computadores**.

Trabajo Fin de Grado «**Herramienta de desarrollo de aplicaciones dinámicamente reconfigurables en FPGA**», disponible en <http://hdl.handle.net/10578/12273>.
Calificación: Matrícula de honor.

experiencia profesional

Feb 2018
Actualmente

TECNOBIT (GRUPO OESÍA)
Ingeniero Software Junior

Valdepeñas, Ciudad Real

Ingeniería de pruebas - Optrónica

- Diseño, implementación e integración de componentes SW para Equipos de Pruebas Automáticos (ATE).
- Implementación e integración de secuencias de pruebas automáticas para la verificación y calificación de equipos de optrónica.
- Desarrollo y mantenimiento de aplicaciones para el control manual de sistemas.
- Integración continua de componentes SW.

Mar 2017
Ene 2018

TECNOBIT (GRUPO OESÍA)
Becario - Ingeniero Software

Valdepeñas, Ciudad Real

Programa «Escuela Oesía». Funciones realizadas:

- Desarrollo en C++, C#.
- Automatización de pruebas a interfaces gráficas haciendo uso de la plataforma TestComplete.
- Desarrollo de pruebas unitarias utilizando el framework MSTest.
- Integración continua de componentes SW (Jenkins).

habilidades técnicas

Programación C#, C/C++, Java, Python

GUI .NET, Qt

Scripting Shell script (GNU/Linux), PowerShell (Windows), Make, TCL

Testing MSTest, TestComplete (GUI testing)

Hardware Xilinx Vivado, VHDL

Modelado	UML
VCS	SVN, Mercurial, Git
BBDD	SQL
Otras	MPI, Trac, Jenkins, Android development, L ^A T _E X

cursos complementarios

Fundamentos de diseño SW. 4+1 Vistas.

Formación Interna Oesía

Conjunto de principios, leyes y otros que constituyen las bases para el correcto análisis y diseño orientado a objetos:

- DRY, YAGNI, KISS, SOLID, GRASP, ...
- Ley del Cambio Contínuo, de la Mínima Sorpresa, ...
- Técnica de doble despacho, Diseño por Contrato de Meyer, ...

Todo ello junto con *Code smell* (jerarquías de herencia paralela, envidia de características, etc.) para reconocer posibles errores software.

Profesor: Luis Fernández Muñoz. Duración: 25 horas.