



# QA & Testing

## Junior Academy

160 horas

# Práctica final

## BDD, Cucumber y Java

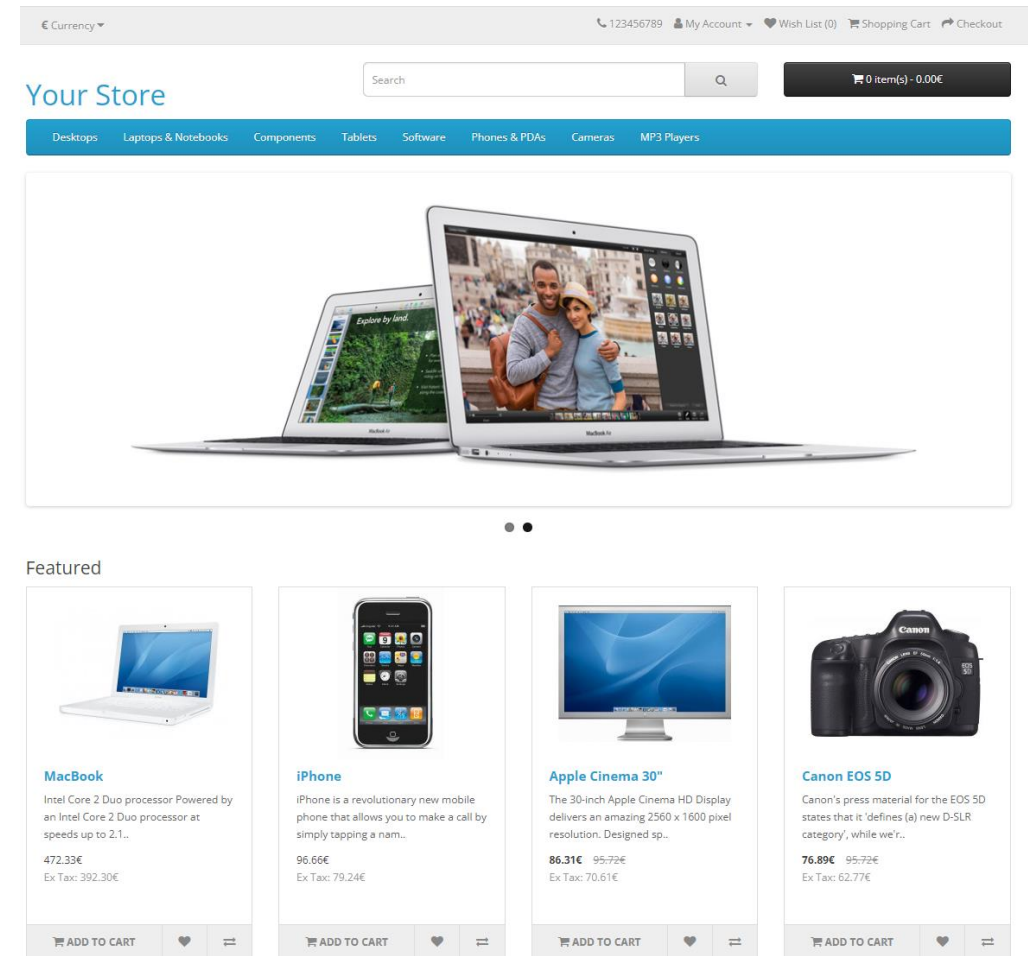
Práctica final (4 días)



La empresa **openCart Abstracta** ha contratado a un equipo de QA para definir, implementar y reportar un plan de pruebas de regresión de su página web: <http://opencart.abstracta.us/>

El **objetivo** es tener un plan de pruebas automatizado que se pueda lanzar cada vez que se genere una versión de la misma o cada vez que el cliente solicite la ejecución de dichas pruebas:

- Definir un plan de pruebas
- Implementar un proyecto automatizado con dicho plan de pruebas
- Establecer un sistema de reporte para ofrecer el resultado de las ejecuciones



# Práctica final

## BDD, Cucumber y Java

Práctica final (4 días)



### Requisitos :

- Definir un plan de pruebas con la mayor cobertura posible de la página web. Tiene que cubrir al menos los casos críticos/básicos de la plataforma:
  - Registro
  - Login
  - Añadir producto al carrito
  - Checkout y compra de un producto
- Implementar un proyecto de pruebas automatizado que sea:
  - Mantenible
  - Legible
  - Reusable
- Definir un sistema de reporte para la revisión de los resultados de forma legible:
  - Añadir capturas de pantalla en caso de error
- Lanzamiento en diferentes navegadores a través de línea de comandos (con modo *headless*)
- Establecer un sistema que permita lanzar de forma sencilla los *smoke test* de la aplicación

# Práctica final

## BDD, Cucumber y Java

Práctica final (4 días)

### Entrega :

- Se deberá subir una rama al proyecto <https://github.com/ramonlr93/HiberusUniversitySelenium>
- El nombre de la rama será *openCart\_nombre\_apellido*, ejemplo:
  - *openCart\_jonatan\_villen*
- Se deberá poder lanzar con el siguiente comando:
  - `mvn clean test`
  - `mvn clean test -Dcucumber.filter.tags="@smoke"`
- Se valorará el uso de todo lo visto durante el curso:
  - Selenium
  - BDD
  - Cucumber
  - Gherkin
  - Reporting
  - Patrones de diseño
- **Presentación de la práctica (30 minutos máximo)**
- **Extra:**
  - Paralelización de la ejecución de los casos de prueba
  - Ejecución en un contenedor Docker
  - Integración continua CI / CD (Jenkins)