

Mis libros favoritos de Desarrollo de Software

Hay un porcentaje muy pequeño de libros que merezcan la pena de verdad. En los últimos 20 años me habré leído más de un centenar de libros. Sin embargo, que merezcan la pena de verdad, solo un 10%.

Ese 10% me los habré releído una decena de veces.

Si hubiera sabido qué libros tenía que leerme cuando empecé, me habría ahorrado mucho tiempo y dinero. Echa cálculos.

Hay personas que consumen una cantidad ingente de libros, pero no avanzan demasiado.

Porque no te harás un experto leyendo 100 libros de desarrollo de software. Pero estarás mucho más cerca leyéndote 10 veces los 10 mejores libros de desarrollo de software.

Empecé con esta recopilación hace 20 años.

En esa lista solo hay libros buenos, de esos que cambian una carrera profesional. Con la mía funcionó. Ese 10% que me he leído, releído y realmente merecen la pena.

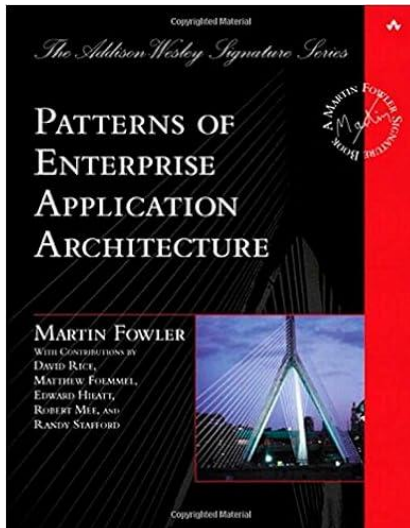
Lo que aprenderás en estos libros no te hará más popular en las redes sociales.

Pero una vez tengas la experiencia adecuada en lo que enseñan, no te faltará nunca trabajo y verás aumentados tus ingresos considerablemente.

Son una parte de inspiración de mi newsletter y en mis cursos, además de mi experiencia.

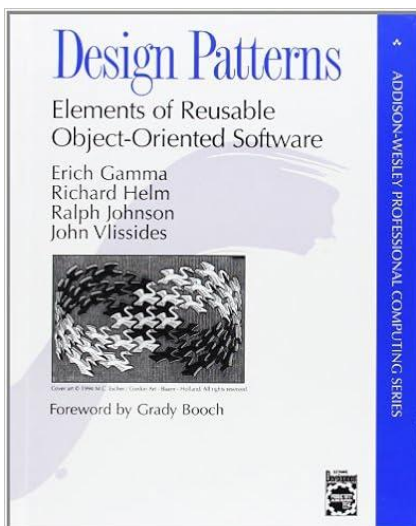
¡Disfrútala!

Diseño y Arquitectura



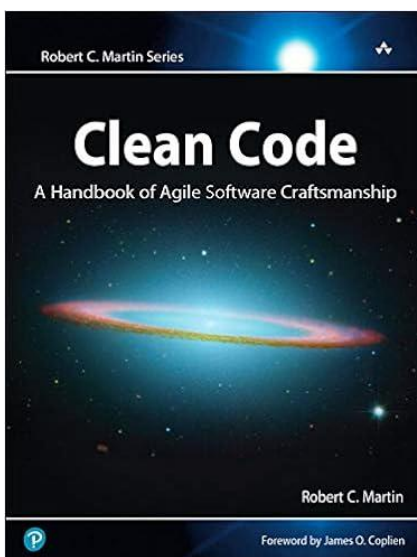
Patterns of Enterprise Application Architecture de Martin Fowler

Martin Fowler define las ideas y técnicas centrales que cientos de miles de desarrolladores han utilizado para mejorar su software.



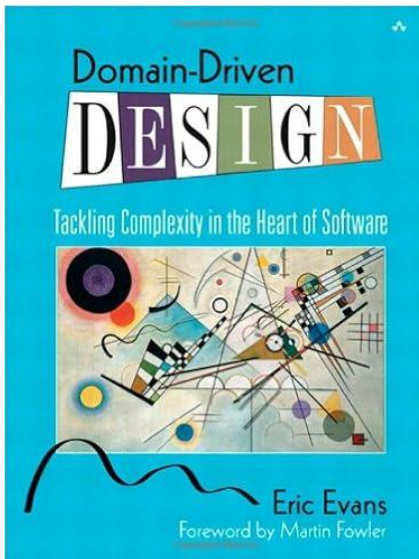
Design Patterns: Elements of Reusable Object-Oriented Software de Erich Gamma

Con una gran experiencia en el diseño de software orientado a objetos, cuatro diseñadores de primer nivel presentan un catálogo de soluciones simples y concisas para problemas de diseño comunes.



Clean Code: A Handbook of Agile Software Craftsmanship de Robert C. Martin

Robert C. Martin nos presenta en este libro los principios básicos del código limpio.

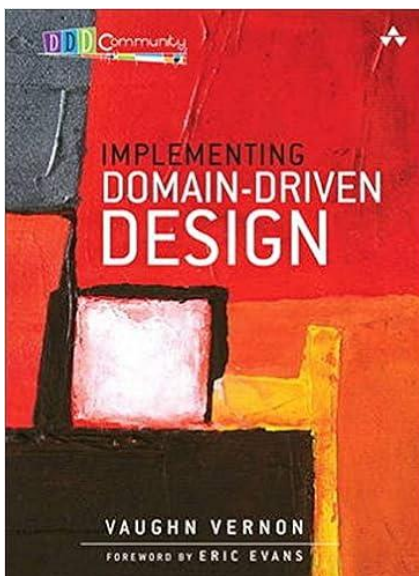


Domain-Driven Design: Tackling Complexity in the Heart of Software

de Eric J. Evans

Primer libro sobre desarrollo orientado al dominio, donde Eric J. Evans introduce conceptos que hoy en día son muy comunes en DDD.

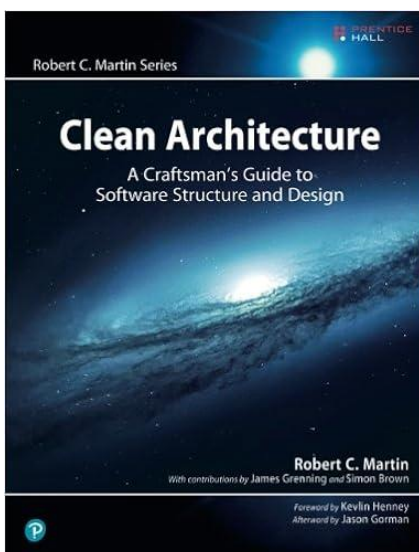
Defiende de manera convincente la importancia del modelado de dominios como el enfoque central del desarrollo y proporciona un marco sólido y un conjunto de técnicas para lograrlo.



Implementing Domain-Driven Design

de Vaughn Vernon

Vaughn Vernon combina enfoques guiados para la implementación con arquitecturas modernas, destacando la importancia y el valor de centrarse en el dominio empresarial.

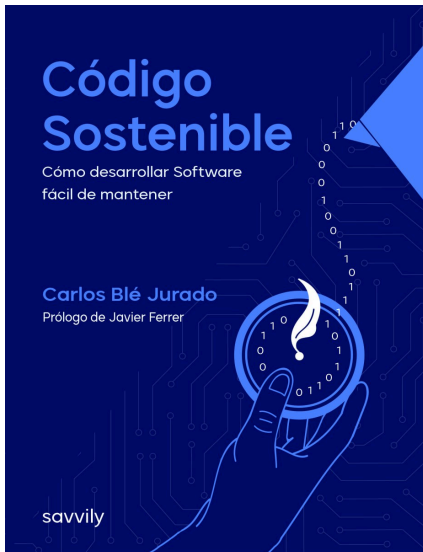


Clean Architecture: A Craftsman's Guide to Software Structure and Design

de Robert C. Martin

Robert C. Martin nos da a conocer Clean Architecture, consiste en una serie de reglas y recomendaciones adquiridas a lo largo de su extensa carrera, nos enseña a desarrollar software mantenible y escalable.

Continuando así el éxito sus libros más vendidos, "Clean Code" y "The Clean Coder".

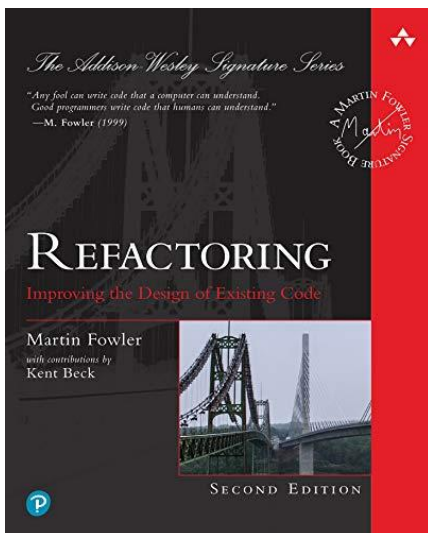


Código Sostenible

de Carlos Blé Jurado

Cómo programar código fácil de mantener. El libro que nos hubiera gustado tener entre nuestras manos cuando estábamos aprendiendo a programar. Una guía para quienes buscan la satisfacción del código bien hecho. ¿Te has planteado alguna vez cómo sería programar sin prisas, sin parches y sin chapuzas? Tras veinte años de carrera como programador, consultor, conferenciante y formador, Carlos Blé plasma lo mejor de sus enseñanzas en este libro.

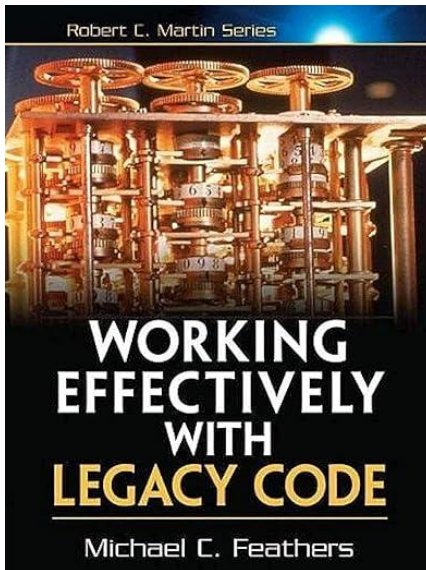
Código legado



Refactoring: Improving the Design of Existing Code

de Fowler Martin

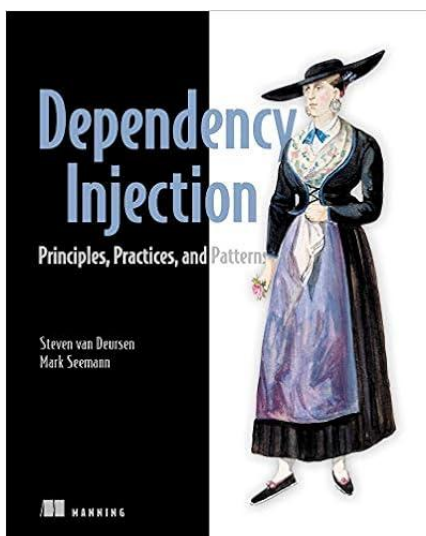
Segunda edición del clásico Refactoring de Martin Fowler, donde muestra diferentes técnicas de refactoring para trabajar con código legado.



Refactoring: Improving the Design of Existing Code de Michael C. Feathers

Este libro proporciona a los programadores la capacidad de manejar de manera rentable problemas comunes de código heredado sin tener que pasar por la tarea enormemente costosa de reescribir todo el código existente. Sus estrategias son oro molido.

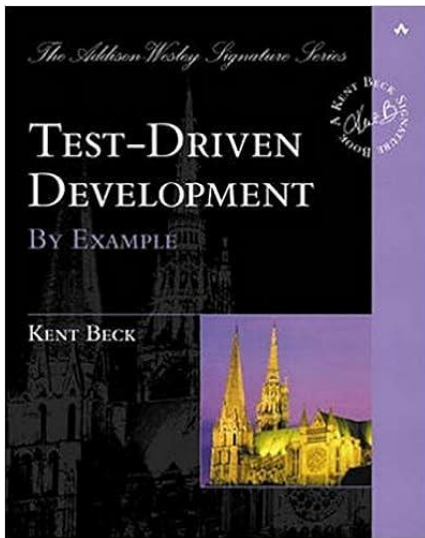
Inyección de dependencias



Dependency Injection Principles, Practices and Patterns de Mark Seemann

La inyección de dependencia se ha convertido en una práctica común en la última década. Esta segunda edición enseña a los lectores a utilizar Dependency Injection para reducir las dependencias entre los componentes de la aplicación. Los lectores se sumergirán en ejemplos detallados, Patrones DI y anti-patrones.

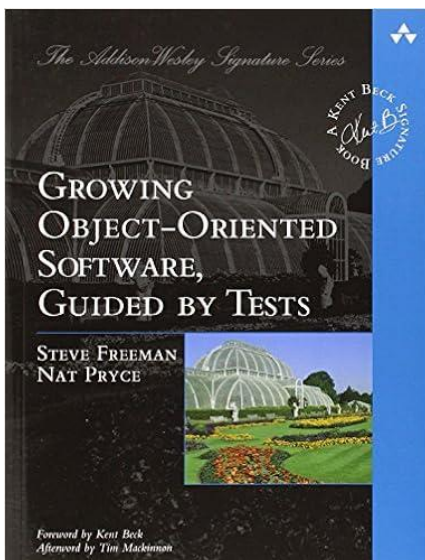
Testing



Test Driven Development: By Example

de [Kent Beck](#)

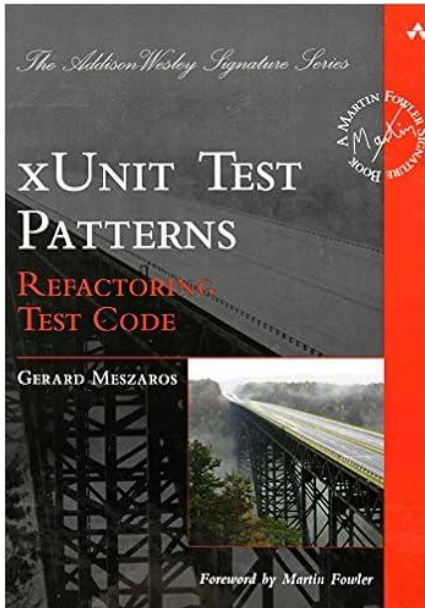
Los desarrolladores enfrentan desafíos de programación complejos todos los días, sin embargo, no siempre están preparados para determinar la mejor solución. Para reunir la fuerza y el coraje necesarios para superar tareas aparentemente hercúleas, los programadores deben buscar el desarrollo basado en pruebas (TDD), un conjunto probado de técnicas que fomentan diseños simples y conjuntos de pruebas que inspiran confianza.



Growing Object-Oriented Software, Guided by Tests

de [Steve Freeman](#) , [Nat Pryce](#)

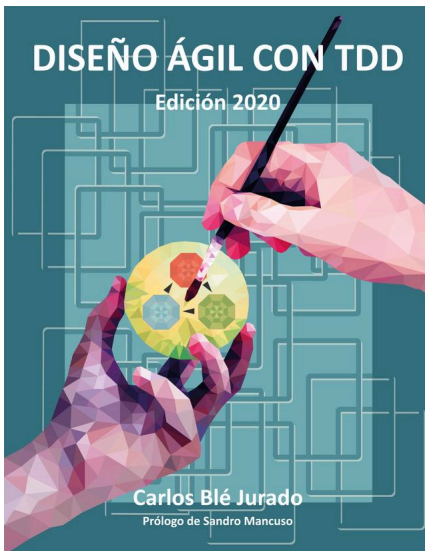
Steve Freeman y Nat Pryce describen los procesos que utilizan, los principios de diseño que se esfuerzan por lograr y algunas de las herramientas que les ayudan a realizar el trabajo. A través de un ejemplo trabajado extendido, aprenderá cómo funciona TDD en múltiples niveles, usando pruebas para impulsar las características y la estructura orientada a objetos del código, y usando Mock Objects para descubrir y luego describir relaciones entre objetos.



xUnit Test Patterns: Refactoring Test Code

de Gerard Meszaros

Las pruebas automatizadas son la piedra angular del desarrollo ágil. Una estrategia de prueba eficaz ofrecerá nuevas funciones de forma más agresiva, acelerará los comentarios de los usuarios y mejorará la calidad. Sin embargo, para muchos desarrolladores, la creación de pruebas automatizadas efectivas es un desafío único y desconocido. xUnit Test Patterns es la guía definitiva para escribir pruebas automatizadas utilizando xUnit, el marco de pruebas unitarias más popular en uso en la actualidad.

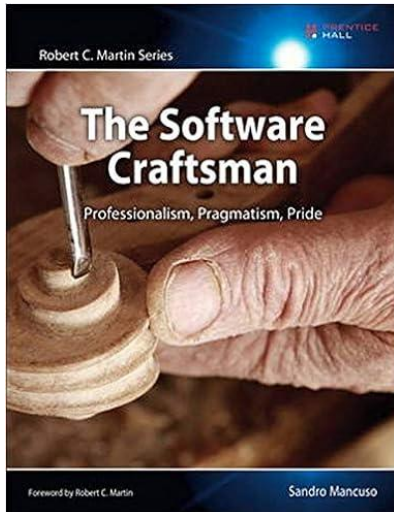


Diseño Ágil con TDD

de Carlos Blé Jurado

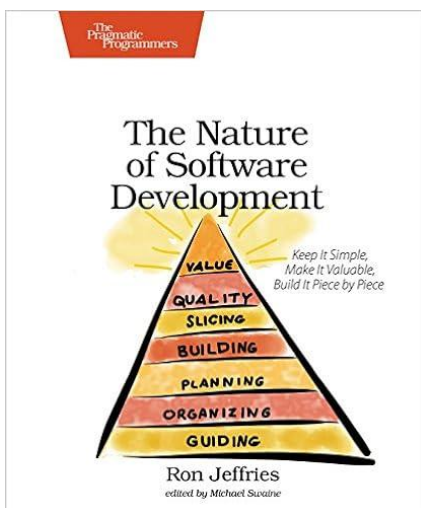
Primer libro en castellano sobre Test-Driven Development. La primera edición se lanzó en enero de 2010. Esta es una edición totalmente nueva, actualizada, más práctica, más fácil de leer. Al grano, para aquellas personas que quieren empezar a practicar TDD, mejorar la técnica o quieren aprender sobre automatización de pruebas de código.

Desarrollo de software



The Software Craftsman: Professionalism, Pragmatism de Sandro Mancuso

Sea un mejor desarrollador y proporcione un mejor código. A pesar de las herramientas y metodologías avanzadas, los proyectos de software siguen fallando. ¿Por qué? Demasiadas organizaciones todavía ven el desarrollo de software como una línea de producción más. Demasiados desarrolladores también se sienten así, y se comportan en consecuencia. En *The Software Craftsman: Professionalism, Pragmatism, Pride*, Sandro Mancuso ofrece un camino mejor y más satisfactorio.



The Nature of Software Development: Keep It Simple, Make It Valuable, Build It Piece by Piece de Ron Jeffries

Necesita obtener valor de su proyecto de software. Lo necesita "gratis, ahora y perfecto". No podemos llevarlo allí, pero podemos ayudarlo a que sea "más barato, más rápido y mejor". Este libro lo lleva desde el deseo de obtener valor hasta las actividades específicas que ayudan a los buenos proyectos ágiles a entregar mejor software antes y a un costo menor. Utilizando simples bocetos y pocas palabras, el autor te invita a seguir su camino de aprendizaje y comprensión desde medio siglo de desarrollo de software y desde su compromiso con los métodos ágiles desde sus inicios.