|  |  |  |
| --- | --- | --- |
| **Code Smell** | **Donde se Encuentra (class.method)** | **Refactoring** |
| Long Method | Order.Total | Extract Method |
| Primitive Obsession | ThirdParty.FormatPhoneNumber | Extract Class |
| Long Parameter List | Order.\_constructor | Introduce Parameter Object |
| Dataclums | Variables street, city, country en Order y Employee | Extract Class |
| Switch Statements | Order.Total | Replace Conditional with Polymorphism |
| Refused Bequest | Herencia Employee -> Salesman  (.Subordinates, .AddSubordinate, .RemoveSubordinate) | Push Down Field/Method |
| Divergent Change | Product.ToXml | Extract Class |
| Lazy Class | ImageInfo | Inline Class |
| Duplicate Code | Manager.NetSalary, Salesman.NetSalary | Pull Up Field/Method |
| Speculative Generality | ThirdPary | Collapse Hierarchy |
| Feature Envy | Order.CalculateItemAmount | Move Method |
| Inappropriate Intimacy | Order.OrderItems | Encapsulate Collection |
| Comments | * Fields, Inside/Outside Methods (Debido a la falta de claridad o nombres que no revelan el propósito real): Salesman, Order.Total * Obvios (Comentarios que no dan ninguna información adicional): OrderItem.\_create, Order.Total. * Scary Noise(Ruidosos): Product fields. * Mandatorios (No existe una regla universal que diga que todos los métodos deben tener un javadoc, o cada parámetro debe tener un comentario. El mejor contexto para utilizar estos javadoc es en frameworks o libreríos, no tanto en aplicaciones). Este tipo de comentarios se prestan a la confusión general, desorganización, mentira y ocultan el código real.) Salesman.UpdateQuota | Extract Method  Rename |