

# ESBOZO DE IMPLEMENTACIÓN RUST – GOODWARE

Autor: Wolfspell & Collaborative AIs | Idioma: es | Fecha: 2025-07-06

```
use analytics_crate;
use pdf_report_generator;

// Recolector de datos
struct DataIngestor;
impl DataIngestor {
    fn collect_data(&self) -> Data {
        println!("Recolectando datos de producción...");
        Data {}
    }
}

// Analizador de emisiones
struct EmissionAnalyzer;
impl EmissionAnalyzer {
    fn analyze(&self, data: &Data) -> AnalysisResult {
        println!("Analizando emisiones...");
        AnalysisResult {}
    }
}

// Optimizador de proceso
struct ProcessOptimizer;
impl ProcessOptimizer {
    fn optimize(&self, analysis: &AnalysisResult) -> Recommendations {
        println!("Generando recomendaciones...");
        Recommendations {}
    }
}

struct Data {}
struct AnalysisResult {}
struct Recommendations {}
```

```
fn main() {  
    println!("Verificando seguridad y conformidad...");  
    let ingestor = DataIngestor;  
    let analyzer = EmissionAnalyzer;  
    let optimizer = ProcessOptimizer;  
  
    let data = ingestor.collect_data();  
    let analysis = analyzer.analyze(&data);  
    let recommendations = optimizer.optimize(&analysis);  
  
    println!("Generando informe PDF semanal...");  
    // pdf_report_generator::generate(recommendations, "reporte.pdf");  
}
```

*Este código es un boceto; debe ser auditado por especialistas antes de uso operativo. Utiliza tipado fuerte, evita 'unsafe' y se integrará con logs inmutables y quorum ético según el manifiesto.*