

Ahora complementemos la seccion de pkg/reports complementando con les reportes mejorados del proyecto 1 generando de mejor manera la reporteria.

- ¡Perfecto! Voy a implementar el sistema de reportes completo basándome en el Proyecto 1. Primero analizo qué reportes tiene:

- Bash(find /home/julian/Documents/MIA_2S2025_P1_201905884/backend -type f -name ".go" -path "/Rep/*" | head -10)
└─ /home/julian/Documents/MIA_2S2025_P1_201905884/backend/command/Rep/repo
rtes.go
/home/julian/Documents/MIA_2S2025_P1_201905884/backend/command/Rep/opti
... +13 lines (ctrl+o to expand)

- Read(~/.Documents/MIA_2S2025_P1_201905884/backend/command/Rep/reportes.go)
└─ Read 100 lines (ctrl+o to expand)

- Voy a crear la estructura de reportes en pkg/reports. Primero creo el archivo principal:

- Bash(mkdir -p
/home/julian/Documents/MIA_2S2025_P2_201905884/Backend/pkg/reports)
└─ (No content)

- Write(pkg/reports/reports.go)
└─ Error writing file

- Bash(ls /home/julian/Documents/MIA_2S2025_P2_201905884/Backend/pkg/reports/
2>/dev/null || echo "Directory empty or doesn't exist")
└─ disk_reports.go
dot_builder.go
fs_reports.go
... +3 lines (ctrl+o to expand)

PROF

- Ya existe un pkg/reports! Voy a revisar qué hay:

- Read(pkg/reports/reports.go)
└─ Read 79 lines (ctrl+o to expand)

- Perfecto! Ya hay una base. Voy a revisar los demás archivos:

- Read(pkg/reports/disk_reports.go)
└─ Read 87 lines (ctrl+o to expand)

- Excelente! Ya hay generadores de DOT. Ahora voy a crear los adaptadores que lean del disco real y generen imágenes. Primero voy a leer el P1 para ver cómo lee los datos:

- Read(~/.Documents/MIA_2S2025_P1_201905884/backend/command/Rep/reportes.go)
└─ Read 80 lines (ctrl+o to expand)

- Ahora voy a crear los generadores que conecten con el disco real. Primero, un archivo para los generadores principales:

- Write(pkg/reports/generators.go)

- └ Wrote 302 lines to pkg/reports/generators.go

```
package reports
```

```
import (
```

```
"context"
```

```
"fmt"
```

```
"os"
```

```
"os/exec"
```

```
"path/filepath"
```

```
"strings"
```

```
"time"
```

```
... +292 lines (ctrl+o to expand)
```

- Ahora necesito agregar el generador de Superblock al archivo de reportes. Primero leo fs_reports.go para ver qué hay:

- Read(pkg/reports/fs_reports.go)

- └ Read 97 lines (ctrl+o to expand)

- Perfecto! Ya existe ReportSuperBlock. Solo necesito crear un alias:

- Update(pkg/reports/fs_reports.go)

- └ Updated pkg/reports/fs_reports.go with 5 additions

```
72 return d.close()
```

```
73 }
```

```
74
```

```
75 + // ReportSuperblock es un alias para ReportSuperBlock
```

```
76 + func ReportSuperblock(sb SuperBlock, opt Options) string {
```

```
77 + return ReportSuperBlock(sb, opt)
```

```
78 + }
```

```
79 +
```

```
80 // ReportSuperBlock lista campos clave del superblock.
```

```
81 func ReportSuperBlock(sb SuperBlock, opt Options) string {
```

```
82 d := newDot(or(opt.Title, "SuperBlock"))
```

- Ahora voy a crear un comando para ejecutar reportes. Primero actualizo el todo:

- Ahora voy a crear un comando REP en el paquete de comandos:

- Write(internal/commands/rep.go)

- └ Wrote 116 lines to internal/commands/rep.go

```
package commands
```

```
import (
```

```
"context"
```

```
"fmt"
"strings"
"MIA_2S2025_P2_201905884/pkg/reports"
)
... +106 lines (ctrl+o to expand)
```

• Ahora necesito agregar CmdRep a types.go y actualizar el parser:

• Read(internal/commands/types.go)

└ Read 50 lines (ctrl+o to expand)

• Update(internal/commands/types.go)

└ Updated internal/commands/types.go with 3 additions

```
35 CmdJournaling CommandName = "journaling"
```

```
36 CmdRecovery CommandName = "recovery"
```

```
37 CmdLoss CommandName = "loss"
```

```
38 +
```

```
39 + // Comandos de reportes
```

```
40 + CmdRep CommandName = "rep"
```

```
41 )
```

```
42
```

```
43 // CommandHandler es la interfaz que implementan todos los
handlers de comandos
```

• Ahora actualizo el parser para incluir REP:

• Read(internal/commands/parser.go)

└ Read 80 lines (ctrl+o to expand)

• Update(internal/commands/parser.go)

└ Updated internal/commands/parser.go with 2 additions

```
59 return parseRecovery(args)
```

```
60 case CmdLoss:
```

```
61 return parseLoss(args)
```

```
62 + case CmdRep:
```

```
63 + return parseRep(args)
```

```
64 default:
```

```
65 return nil, fmt.Errorf("comando desconocido: %s", cmd)
```

```
66 }
```

└ Claude Code v2.0.8

| | Tips for getting started |

| Welcome back Santiago! | Run /init to create a CL... |

| | _____ |

|  | Recent activity |

|  | 1h ago Basados en todo ... |

| ■ ■ ■ | /resume for more |

Sonnet 4.5 · Claude Pro	
/.../Documents/MIA_2S2025_P2_201905884/Backend	

===== Conversation compacted · ctrl+o for history =====

- └ Read internal/commands/parser.go (347 lines)
- └ Read internal/commands/types.go (417 lines)
- └ Read internal/commands/rep.go (116 lines)
- └ Read pkg/reports/fs_reports.go (102 lines)
- └ Read pkg/reports/generators.go (302 lines)
- └ Todo list read (6 items)
- └ Session limit reached · resets 6pm

/upgrade to increase your usage limit.