

Camada Domain – Descrição Completa dos Arquivos

Estrutura

```
Domain/  
├── Entities/  
│   └── Task.swift  
├── UseCases/  
│   ├── GetTasksUseCase.swift  
│   ├── AddTaskUseCase.swift  
│   ├── UpdateTaskUseCase.swift  
│   └── DeleteTaskUseCase.swift  
└── Repositories/  
    └── TaskRepository.swift
```

Domain/Entities/Task.swift

Função

Representar o modelo de domínio do app: uma **tarefa**.

Essa entidade é usada em todas as camadas, mas **não depende de nenhuma delas**.

Conteúdo

```
struct Task: Identifiable, Equatable {  
    let id: UUID  
    var title: String  
    var isCompleted: Bool  
}
```

Conexões

- Usada pelos UseCases, Repositories, ViewModel e View
- Representa um objeto puro de negócio (sem lógica de UI ou persistência)



Domain/Repositories/TaskRepository.swift



Função

Definir o **contrato de acesso a dados** que será implementado na camada de dados.

A interface que os UseCases conhecem e utilizam.



Conteúdo

```
protocol TaskRepository {  
    func fetchTasks() -> [Task]  
    func addTask(_ task: Task)  
    func updateTask(_ task: Task)  
    func deleteTask(_ task: Task)  
}
```



Conexões

- Usado por todos os UseCases
- Implementado na camada Data (TaskRepositoryImpl)
- É injetado nos UseCases via inicialização



Domain/UseCases/GetTasksUseCase.swift



Função

Caso de uso responsável por **listar todas as tarefas**.



Conteúdo

```
struct GetTasksUseCase {  
    let repository: TaskRepository  
  
    func execute() -> [Task] {  
        repository.fetchTasks()  
    }  
}
```

Conexões

- Usa `TaskRepository` (protocolo)
 - Retorna `[Task]` para a ViewModel
-



Domain/UseCases/AddTaskUseCase.swift



Função

Adicionar uma nova tarefa no repositório.



Conteúdo

```
struct AddTaskUseCase {  
    let repository: TaskRepository  
  
    func execute(task: Task) {  
        repository.addTask(task)  
    }  
}
```



Domain/UseCases/UpdateTaskUseCase.swift



Função

Atualiza o status ou conteúdo de uma tarefa.



Conteúdo

```
struct UpdateTaskUseCase {  
    let repository: TaskRepository  
  
    func execute(task: Task) {  
        repository.updateTask(task)  
    }  
}
```



Domain/UseCases/DeleteTaskUseCase.swift



Função

Remove uma tarefa do repositório.



Conteúdo

```
struct DeleteTaskUseCase {  
    let repository: TaskRepository  
  
    func execute(task: Task) {  
        repository.deleteTask(task)  
    }  
}
```



Resumo

Arquivo	Função	Conhece quem?
Task.swift	Representa a entidade de domínio do app	Todas as camadas usam
TaskRepository.swift	Contrato de acesso a dados	UseCases e camada Data
GetTasksUseCase.swift	Lista tarefas	TaskRepository
AddTaskUseCase.swift	Adiciona tarefa	TaskRepository
UpdateTaskUseCase.swift	Atualiza tarefa	TaskRepository
DeleteTaskUseCase.swift	Deleta tarefa	TaskRepository