

Roles y Patrones aplicados

REFERENCIA DE ROLES

- **Information holder:** Conoce y brinda información
- **Structurer:** Mantiene relaciones e información entre objetos.
- **Service Provider:** Realiza trabajo y generalmente ofrece servicios de cómputo.
- **Coordinator:** Reacciona a eventos delegando tareas a otras clases
- **Controller:** Toma decisiones y direcciona las acciones de otras clases
- **Interfacer:** transforma información y pedidos entre distintas partes de nuestro sistema/programa

CLASIFICACION DE CLASES

- **TelegramBot:** Interfacer
- **TelegramAdapter:** Interfacer
- **IMessage:** Information Holder
- **Match:** Controller
- **MatchList:** information holder
- **UserList:** information holder
- **User:** Service provider
- **Battle:** Controller
- **CombinelImage:** Service Provider
- **Menu:** Information Holder
- **Conversation:** Service provider
- **Text:** Informacion Holder
- **Board:** Interfacer
- **Ship:** Structurer

PATRONES APLICADOS

- **MatchList:** esta clase implementa el patrón expert, ya que es la experta en conocer todas las partidas activas. También usa el patrón singleton para que haya una o solo una lista de este tipo.
- **UserList:** esta clase implementa el patrón expert, ya que es la experta en conocer todos los usuarios activos. También usa el patrón singleton para que haya una o solo una lista de este tipo.
- **User:** esta clase utiliza SRP, ya que su única responsabilidad es conocer el usuario y sus datos.
- **Match:** La clase match es experta en hacer toda la lógica del programa.