Gerardo Y. Soto Ríos

Lecture Topic Task

Layered Architecture Applied to Tournaments

Why is it useful?

Layered Architecture is a great way to make testing in the project easier. It allows developers to pinpoint where a problem is within the spectrum of the code. If the domain fails, the rest of the layers will likely follow, but each layer can also be tested independently to find the exact issue. This separation allows for a clearer and faster debugging and testing process among developers in the project, making it a very useful structure to use.

Layers of Tournaments

Domain:

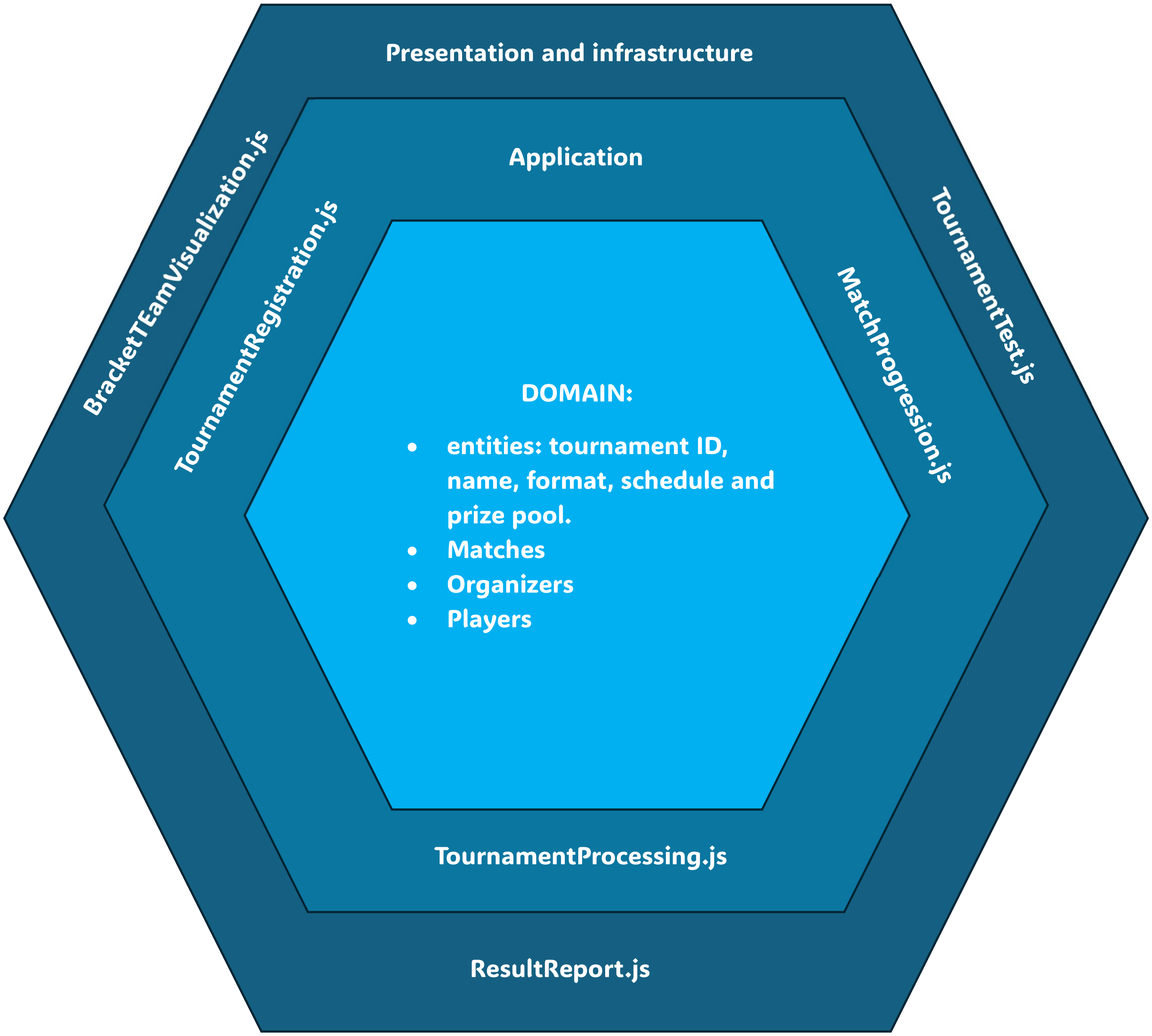
This is where the main entities of the Tournament system are defined, with the tournament itself being the central core. It contains essential attributes such as the tournament ID, name, format, schedule, and prize pool, along with related entities like matches, players, and organizers. The domain layer focuses on establishing how these entities relate to one another and how they function within the tournament context. It defines the core structure that establishes how tournaments, matches, and participants interact.

Application:

This provides the functionality needed to create tournaments, manage player registration, and schedule matches. It ensures that the main processes of a tournament can be properly executed and developed from start to finish. The application layer coordinates these actions, using the entities defined in the domain to carry out each operation in an organized and controlled way. It manages the flow of events such as creating new tournaments, adding players, and setting up the match schedule.

Presentation and infrastructure:

In this section the tournament is displayed and interacted with. It manages the visualization of the tournament bracket and shows the results of each match. It is also where tests are handled and executed to ensure everything functions as intended. In addition, this layer organizes the layout and arrangement of the tournament's data, making it accessible and understandable to users.


Presentation and infrastructure
BracketTEamVisualization.js
TournamentTest.js
Application
TournamentRegistration.js
MatchProgression.js
DOMAIN:
entities: tournament ID, name, format, schedule and prize pool.
Matches
Organizers
Players
TournamentProcessing.js
ResultReport.js