

 Institución Universitaria	ESPECIFICACIONES PARA IMPLEMENTAR FSM NAVAL BATTLE	Tema	04
Asignatura: MATEMÁTICAS PARA LA INFORMÁTICA AVANZADA			
Código-Grupo: 190304003-1		Programa académico: Ingeniería de Sistemas	

Contenido

Consideraciones Generales	1
Especificaciones	2
Componentes del Juego	2
Dinámica del juego	3
Arquitectura del sistema	4
Definiciones de los Autómatas Finitos (FA)	5
Mockup (modo consola)	8

Consideraciones Generales

Todos los equipos diseñarán e implementarán el sistema de FSM con los mismos elementos de comunicación y mensajes.

- Entrega
 - Miércoles 04 de Junio del 2025
 - Presencial en el aula de clases.
- Formato
 - Código fuente completo y documentado para permitir su correcta ejecución.
 - Presentación presencial y revisión del sistema mediante enfrentamiento entre equipos a realizar el día Miércoles 04 de Junio.
- MVP (Minimum Viable Product, equivalente a una calificación de 3.0 puntos)
 - Una aplicación funcional, basada en las especificaciones indicadas para el uso de máquinas de estado finito. Es decir, el sistema debe poder iniciar y establecer comunicación con los sistemas de otros equipos. Para esto se tienen definidos mensajes de prueba, enviar un A0 recibirá una respuesta con código 200 que indica comunicación OK, o también si se envía un ataque con una cuadrícula no existente, por ejemplo, X9, el servidor devolvería una respuesta con un código 404, que también indicaría comunicación OK. Si se envía una cuadrícula válida, recibirá respuesta acorde al resultado.
 - La interfaz gráfica del usuario mínima requerida puede ser a modo consola, si el equipo desea realizar algo más gráfico depende de cada equipo y será valorado como tal. Se presentan sugerencias en modo consola, recordando que el objetivo básico está enfocado en desarrollar la FSM.
 - No se requiere documentación.

- Mínimo se requiere desarrollar un Servidor FSM y un Cliente de Ataque FSM. Los equipos podrían desarrollar sólo alguno de los programas si por temas de tiempo no alcanza a desarrollar ambos, pero entonces eso no garantizarían los 3.0 puntos.
- Porcentaje de evaluación 15%
- Criterios de Evaluación
 - Aplicación de conceptos matemáticos sobre autómatas finitos.
 - Calidad y usabilidad de la interfaz de usuario.
 - Implementación de programas con FSM.
 - Manejo apropiado de la comunicación, basado en los alfabetos, lenguajes definidos.

Especificaciones

Desarrollar un sistema basado en máquinas de estado finito para realizar enfrentamientos entre los equipos con un modelo de juego FSM Naval Battle en un entorno cliente-servidor con sockets.

- **Nombre del Proyecto: FSM Naval Battle**
- **Contexto General:** Cada equipo diseñará e implementará un FSM que representa su sistema de defensa naval. Luego, se realizarán enfrentamientos con otro equipo, donde lanzarán una serie de “ataques” (cadenas de entrada) que simulan misiles buscando destruir la flota enemiga. El FSM actúa como el controlador de defensa, que decide si el ataque impacta o es repelido, según su lógica de estados.

Componentes del Juego

1. Flota como una cuadrícula (oculta al oponente)

- Cada equipo tiene una matriz de 5x5.
- En las casillas distribuirá las 3 embarcaciones:
 - ✓ Un Destructor (Destroyer), ocupará 1 casilla
 - ✓ Un Submarino (Submarine), ocupará 2 casillas
 - ✓ Un Acorazado (Battleship), ocupará 3 casillas
- Las posiciones se definen al inicio y no cambian durante el juego.

	1	2	3	4	5	
A						
B						
C						
D						
E						

Destroyers 1 casilla
Submarine 2 casillas
Battleships 3 casillas

2. Misiles como cadenas de salida/entrada

- Cada ataque del oponente es una cadena.
- Filas: A,B,C,D,E y Columnas:1,2,3,4,5.
- Por ejemplo, luego de 5 ataques, este sería el mapa resultante:
 - ✓ “A1” → casilla A1: Fila A, Col 1,
 - ✓ “A5” → casilla A5: Fila A, Col 5,
 - ✓ “B2” → casilla B2,
 - ✓ “C2” → casilla C2,
 - ✓ “E4” → casilla E4

	1	2	3	4	5	
A	X				X	
B		X				
C			X			
D						
E				X	X	

3. FSM como sistema de defensa

- Cada equipo diseñará un autómata que recibe las cadenas de entrada (los misiles).

- El FSM:
 - ✓ ACEPTE una cadena, si el misil golpea una parte de su flota.
 - ✓ RECHAZA si es agua o si esa parte ya fue destruida antes.
- Transiciones reflejan el cambio de estado del tablero: fallido (miss), impactado (hit), hundido (sunk).

	1	2	3	4	5	
A						Fallido
B						
C						Impactado
D						
E						Hundido

Dinámica del juego

1. Fase de diseño:

- Cuadrícula de barcos (puede ser manual o en código).
- Estados como:
 - ✓ "Flota intacta"
 - ✓ "Barco parcialmente dañado"
 - ✓ "Submarino destruido"
 - ✓ "Victoria enemiga", etc.
- Transiciones activadas por entradas como "B2" (misil).

2. Fase de ataque

- Un equipo lanza cadenas (misiles).
- El FSM del defensor procesa esas entradas y responde:
 - ✓ "Impactado" (si acepta la cadena).
 - ✓ "Fallido" (si no la acepta).
 - ✓ "Hundido" (si el FSM cambia a estado de hundimiento).
- El sistema llevará el conteo de los impactos y mostrará el avance.

3. Final del juego

- Gana quien hunda primero toda la flota del otro.

Ejemplo de FSM para un barco en B2 y C2

Estados:

- S0: barco intacto
- S1: barco parcialmente dañado (C2)
- S2: barco hundido (B2 y C2 impactados)

Transiciones:

- Entrada "C2" en S0 => S1
- Entrada "B2" en S1 => S2
- Entrada cualquier otra => S0 (sin efecto)

	1	2	3	4	5	
A	X				X	
B		X				
C			X			
D						
E				X	X	

Arquitectura del sistema

Dos programas: Un Servidor y un Cliente

1. Cada equipo corre un programa que actúa como “servidor de defensa”:

- Este programa debe mostrar:
 - ✓ El identificador del equipo, dirección IP y puerto de escucha
 - ✓ La cuadrícula con la flota definida (modo texto o visual).
 - ✓ El FSM (modelo de defensa) que procesa los ataques recibidos.
- Escuchará en una IP y puerto específicos, esperando recibir cadenas (ataques).
- Cuando recibe un ataque:
 - ✓ Procesa la entrada con su FSM. Devuelve la respuesta a la cadena del ataque: rechazado:fallido, aceptado:impactado, aceptado:hundido / “404:fallido”, “202:impactado”, “200:hundido”
 - ✓ Actualiza su cuadrícula con los estados.

2. El equipo atacante corre su propio programa cliente

- Este programa permite al equipo:
 - ✓ Escribir la IP y puerto del contrincante.
 - ✓ Enviar una cadena (coordenada del misil).
 - ✓ Recibir la respuesta del FSM enemigo.
 - ✓ Mostrar el resultado al jugador (modo texto o visual).

3. Lenguaje de Programación a elección del equipo.

- Opcionalmente, pueden usar Python socket para abrir conexiones TCP entre cliente y servidor. El equipo decidirá cual plataforma o lenguaje utilizará para desarrollar sus aplicaciones.
- El protocolo es simple:
 - El Cliente envía una cadena ("B4", "D1", etc.) a un destino específico.
 - El Servidor la procesa con su FSM y responde un mensaje tipo texto
 - ✓ “rechazado:fallido” / “404-fallido”
 - ✓ “aceptado:impactado” / “202-impactado”
 - ✓ “aceptado:hundido” / “200-hundido”

4. Programas.

- Por cada equipo tendrán dos componentes ejecutables:

1. Servidor de Defensa (FSM)
 - Recibe ataques (coordenadas) de otros equipos.
 - Procesa con su FSM y responde Aceptado o Rechazado, si fue:
 - ✓ 404-fallido,
 - ✓ 202-impactado,
 - ✓ 200-hundido,
 - Define la flota y maneja el estado de cada embarcación.

2. Cliente de Ataque (FSM)

- Permite enviar coordenadas de ataque al servidor del equipo contrario.
- Recibe la respuesta del servidor enemigo.
- Puede tener una interfaz sencilla para registrar los resultados.

Definiciones de los Autómatas Finitos (FA)

A continuación, la especificación completa de un Autómata Finito Determinista (DFA) que modela el comportamiento de un tablero naval que responde a ataques.

Quíntupla del autómata:

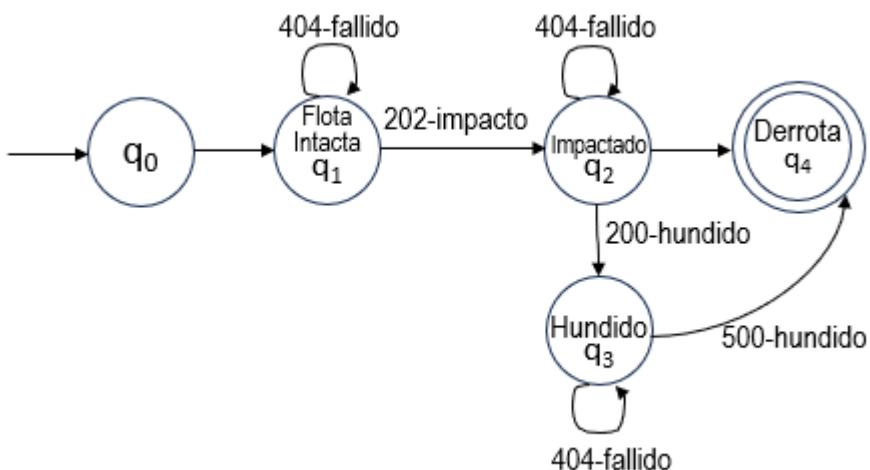
$$M = (Q, \Sigma, \delta, q_0, F)$$

- **Q** (estados): $\{q_0, q_1, q_2, q_3, q_4\}$
- **Σ** (alfabeto de entrada): $\{404\text{-fallido}, 202\text{-impacto}, 200\text{-hundido}, 500\text{-hundido}\}$
- **q_0** (estado inicial): q_0
- **F** (estado final): $\{q_4\}$ (cuando toda la flota ha sido destruida)
- **δ** (función de transición):

Estado Actual	Entrada	Nuevo Estado
q_0	Flota colocada → automático	q_1
q_1	404-fallido (agua)	q_1
q_1	202-impacto	q_2
q_2	404-fallido (agua)	q_2
q_2	200-hundido	q_3
q_3	404-fallido (agua)	q_3
q_3	202-impacto	q_2 (aún queda flota)
q_3	500-hundido (último barco)	q_4 (no queda flota)

Estado Actual	Entrada	Nuevo Estado
q_0	Flota colocada → automático	q_1
q_1	404-fallido (agua)	q_1
q_1	202-impacto	q_2
q_2	404-fallido (agua)	q_2
q_2	200-hundido	q_3
q_3	404-fallido (agua)	q_3
q_3	202-impacto	q_2 (aún queda flota)
q_3	500-hundido (último barco)	q_4 (no queda flota)

Diagrama visual de este DFA



1. Especificación del Autómata Finito para el Servidor de Defensa

Q: Conjunto finito de estados

Cada estado representa el estado global de la flota:

- q_0 : Inicio (flota colocada, sin impactos).
- q_1 : Flota_Intacta, automáticos, ningún barco ha sido impactado.
- q_2 : Impactado, algún barco ha sido impactado, pero todos siguen a flote.
- q_3 : Hundido, al menos un barco ha sido hundido, pero aún queda flota.
- q_4 : Derrota, todos los barcos han sido hundidos.

$$Q = \{q_0, q_1, q_2, q_3, q_4\} : \{\text{Inicio, Flota_Intacta, Impactado, Hundido, Derrota}\}$$

Σ : Alfabeto de entrada

Entradas del sistema = coordenadas de ataque recibidas.

Ejemplo:

$$\Sigma = \{A1, A2, \dots, E5\} \text{ (según el tamaño del tablero, aquí } 5 \times 5)$$

Cada símbolo representa una coordenada posible.

	1	2	3	4	5
A					
B					
C					
D					
E					

δ : Función de transición

La función δ toma un estado actual y una coordenada atacada, y devuelve el nuevo estado.

Formalmente:

$$\delta: Q \times \Sigma \rightarrow Q$$

Ejemplo de transiciones:

- $\delta(\text{Flota_Intacta}, B2) \rightarrow \text{Impactado}$, si impacta pero no se hunde.
- $\delta(\text{Impactado}, C2) \rightarrow \text{Hundido}$, si hunde un barco.
- $\delta(\text{Flota_Intacta}, C3) \rightarrow \text{Fallido}$, si es agua o un ataque previo de impacto.
- $\delta(\text{Impactado}, D4) \rightarrow \text{Derrota}$, si con ese ataque se hunden todos los barcos.

Nota: La transición también requiere información del tablero, por lo que el autómata es en la práctica contextualizado, aunque mantenemos la estructura DFA para efectos académicos.

q_0 : Estado inicial

El sistema comienza en el estado:

$$q_0 = \text{Flota_Intacta} \text{ (una vez la flota ha sido colocada)}$$

F: Conjunto de estados de aceptación

Estado(s) de aceptación = fin del juego

$$F = \{\text{Derrota}\}$$

Gramática del lenguaje reconocido por el autómata

El lenguaje $L(M)$ aceptado por el autómata es el conjunto de secuencias de coordenadas que llevan a la destrucción total de la flota.

Ejemplo:

$L = \{B2, B3, C2, D4, E3, E4, E5, \dots\}$ tal que todas las posiciones de la flota hayan sido alcanzadas.

Resumen ilustrado

Entrada (coordenada)

Estado actual

Resultado

Nuevo estado

B2	q1 (Flota_Intacta)	Impacto	q2 (Impactado)
B3	q2 (Impactado)	Hundido (barco pequeño)	q3 (Hundido)
D4	q3 (Hundido)	Hundido (último barco)	q4 (Derrotado)
C3	q1 (Flota_Intacta)	Fallido (Agua)	q1 (Flota_Intacta)
B3	q1 (Flota_Intacta)	Falludo (Agua)	q1 (Flota_Intacta)

Mockup (modo consola)

1. Servidor de Defensa (FSM).

[SERVIDOR DE DEFENSA - FSM]

❖ Configuración inicial:

Ingresar tamaño del tablero (Ej: 5x5): 5x5

Coloque su flota (Ej: B2 B3):

→ Battleship (3 casillas): E3 E4 E5

→ Submarine (2 casillas): B2 C2

→ Destroyer (1 casilla): B4

✓ Flota colocada correctamente.

🎯 Esperando ataques en el puerto 5000...

✉ Ataque recibido: A1

🔍 Resultado: 404:fallido (Agua)

✉ Ataque recibido: B2

🔍 Resultado: 202-impacto

✉ Ataque recibido: C2

🔍 Resultado: 200-hundido (¡Submarino destruido!)

✉ Ataque recibido: B4

🔍 Resultado: 200-hundido (¡Destroyer destruido!)

...

...

...

✉ Ataque recibido: C1

🔍 Resultado: 500-hundido (¡Destroyer destruido!)

🔥 Toda la flota ha sido destruida. Fin del juego.

2. Cliente de Ataque (FSM)

[CLIENTE DE ATAQUE - FSM]

🔗 Ingrese la IP del enemigo: 192.168.1.22

🔗 Ingrese el puerto: 5000

🧠 FSM de ataque iniciado.

Tablero de ataque 5x5:

1 2 3 4 5

A	-	-	-	-	-
B	-	-	-	-	-
C	-	-	-	-	-
D	-	-	-	-	-
E	-	-	-	-	-

🎯 Ingrese coordenada de ataque (Ej: B2): A1

✉️ Enviando...

✗ Respuesta: 404-fallido (Agua)

🎯 Ingrese coordenada de ataque (Ej: B2): B2

✉️ Enviando...

✓ Respuesta: 202-impactado

🎯 Ingrese coordenada de ataque (Ej: B2): C2

✉️ Enviando...

✓ Respuesta: 200-hundido (¡Submarino destruido!)

🎯 Ingrese coordenada de ataque (Ej: B2): B4

✉️ Enviando...

✓ Respuesta: 200-hundido (¡Destroyer destruido!)

...

...

...

🎯 Ingrese coordenada de ataque (Ej: B2): C1

🔍 Resultado: 500-hundido (¡Battleship destruido!)

🏁 ¡Has ganado! Toda la flota enemiga ha sido destruida.