

# Juego de la vida

## problemática

El reto a realizar consiste en resolver **el juego de la vida** usando solo **JavaScript**, este consta de 4 reglas básicas:

- Cualquier célula viva con menos de **2** vecinos vivos muere a causa de **soledad**.
- Cualquier célula viva con más de **3** vecinos vivos muere a causa de **superpoblación**.
- Cualquier célula viva con **2** o **3** vecinos vivos continúa viva en la siguiente generación.
- Cualquier célula muerta con exactamente **3** vecinos vivos se convierte a una célula viva.

## Solución

La problemática se soluciono de la siguiente manera:

### Análisis

- i. Buscar la manera de dibujar un grid **X,Y**.
- ii. Obtener la cantidad de vecinos vivos de una celda.
- iii. Validar la cantidad de vecinos vivos de una celda dependiendo si es **viva** o **muerta**.
- iv. Dibujar el grid **X,Y** con las nuevas celdas.

### Desarrollo

- **Estructura Del Proyecto:**
  - **index.html**: archivo de implementación del juego.
  - **js**:
    - **game.js**: clase para interacción con el usuario.
    - **board.js**: clase para interactuar con **game** y poder efectuar las operaciones del tablero de juego.
  - **css**
    - **main.css**: estilos para el juego.
    - **font-awesome.css** y **font-awesome.min.css**: iconos para el juego.
  - **fonts**: fuentes de font-awesome.

- **Clase game:**

La clase game permite al usuario interactuar con el juego, usando los siguientes métodos:

- **init::** crea el tablero de juego usando una matriz random de elementos **0**, mediante la clase **gameBoard**.
- **start**: inicia el juego usando una matriz random de **1** y **0**, mediante la clase **gameBoard**.
- **stop**: detiene el juego y reinicia el tablero, mediante la clase **gameBoard**.
- **buttonStyles**: propiedades que se le asignan al botón de play dependiendo del estatus del juego.

- **Clase gameBoard:**

La clase gameBoard está diseñada para crear, eliminar, dibujar, etc. elementos del tablero de juego, utilizando los siguientes métodos:

- **createBoard:** método para crear una matriz de elementos **1** y **0** o sólo de elementos **0** según sea el caso.
- **drawGrid:** método para dibujar un tablero **X,Y**.
- **drawCell:** método para dibujar una celda de color **blanco** o **negro** usando **HTML5 Canvas**.
- **createNextGeneration:** método para obtener la nueva generación de celdas **vivas** o **muertas**.
- **getNeighborsAlive:** método para obtener el número de celdas vivas.
- **validateCell:** método para aplicar las reglas del juego a una celda específica.
- **init:** método para iniciar el tablero.
- **update:** método para iniciar el juego.
- **delete:** método para detener el juego.

## Implementación

Mediante el archivo **index.html** se implementa el juego para que el usuario pueda interactuar con este. Para esto hay dos botones uno de **play** para iniciar el juego y **stop** para detener el juego.

## Conclusión

---

Para terminar debo decir que fue un reto bastante interesante ya que tuve aprender algunos conceptos que desconocía y el resultado se puede ver [aquí](#) .