



***Universidad Tecnológica Nacional***

**INFORME TÉCNICO DE LABORATORIO C : ISLA DE  
SUPERVIVENCIA**

**Materia:** Algoritmos y Estructuras de Datos

**Integrantes:**

- HOFSTETTER, Guadalupe María Itatí. Legajo: 28117
- MONTICELLI, Lucas Emanuel. Legajo: 24344
- POSSIEL, Ezequiel Emanuel. Legajo: 26532

***Ciclo lectivo: 2024***

## **LINK EXPLICATIVO YOUTUBE:**

<https://www.youtube.com/watch?v=UPpgEXCMA9I>

## **LINK REPOSITORIO GITHUB FINAL:**

<https://github.com/EzePoss/Actualizador-Precios>

## **INTRODUCCION:**

El presente informe detalla el desarrollo del Laboratorio de C, para el cual se nos fue asignada la consigna de elaborar un videojuego creado que desafía al jugador a sobrevivir en una isla desierta. El jugador debe recolectar recursos, construir refugios, cazar alimentos y enfrentarse a amenazas como animales salvajes y condiciones climáticas adversas, mientras gestiona indicadores de salud, hambre, sed y energía.

El desarrollo de este videojuego requirió no solo habilidades técnicas avanzadas, sino también creatividad, organización y perseverancia. En este informe, explicamos la estructura del proyecto, las herramientas utilizadas, los desafíos superados y las lecciones aprendidas.

## **ESTRUCTURA DEL PROYECTO:**

En primera instancia, ni bien se nos fue asignada la consigna, decidimos ponernos manos a la obra y sin tocar siquiera una línea de código. Decidimos analizar primero, qué alternativas presentábamos para presentar la interfaz de nuestro juego, si íbamos a ejecutarlo mediante la consola o una gráfica. Luego de una investigación en Google, Youtube y consultando que consejo nos daba la IA, debido a la dificultad del juego, llegamos a la conclusión de que la mejor opción para la interfaz era utilizar una gráfica.

El juego fue desarrollado utilizando la librería gráfica SDL2, complementada con extensiones para el manejo de imágenes (SDL\_image), texto (SDL\_ttf) y audio (SDL\_mixer). Estas herramientas permitieron crear una interfaz gráfica inmersiva y un sistema de sonido que mejora la jugabilidad.

Una vez ya teníamos establecido que íbamos a utilizar la gráfica para ejecutar el juego, desarrollamos diferentes ideas de cómo y qué podríamos utilizar como mapa para simular la isla desierta. Para esto, quisimos utilizar la IA para el desarrollo de una imagen de una isla, pero debido a la complejidad que implicaba trabajar con una imagen, para establecer los recursos y demás objetos del juego, descartamos rápidamente la idea. Finalmente, llegamos a la conclusión que la mejor idea para generar un mapa, era establecer una matriz de 27x27, en la que cada elemento de la matriz representa un tile del mapa. Posteriormente para establecer los colores de cada tile, utilizamos SDL\_Color.

```

178 const int forma_isla[27][27] = {
179     {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
180     {0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0},
181     {0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0},
182     {0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0},
183     {0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0},
184     {0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
185     {0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
186     {0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
187     {0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
188     {0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
189     {0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
190     {0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0},
191     {0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0},
192     {0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
193     {0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
194     {0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0},
195     {0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0},
196     {0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0},
197     {0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0},
198     {0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0},
199     {0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0},
200     {0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0},
201     {0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0},
202     {0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0},
203     {0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0},
204     {0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0},
205     {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
206
207 SDL_Color color_agua = {64, 224, 208, 255};
208 SDL_Color color_arena = {238, 214, 175, 255};
209 SDL_Color color_pasto = {34, 139, 34, 255};

```

## FUNCIONES DEL JUEGO:

El código del proyecto está organizado en módulos que manejan aspectos específicos del juego. Entre las funciones más relevantes se encuentran:

- **Gestión del Tiempo y Clima:**
  - actualizar\_ciclo\_dia\_noche: Simula el paso del tiempo con ciclos de día y noche.
  - actualizar\_clima: Implementa cambios climáticos aleatorios entre sol y tormenta, los cuales influyen en la jugabilidad.
- **Recursos y Construcciones:**
  - generar\_recursos: Genera recursos disponibles en el mapa.
  - manejar\_construccion: Permite construir refugios o mejorar el equipo del jugador.
- **Amenazas y Combate:**
  - actualizar\_animales: Controla el movimiento y comportamiento de los animales salvajes.
  - manejar\_combate: Gestiona el enfrentamiento del jugador con las amenazas.
- **Interfaz Gráfica y Estadísticas:**

- `renderizar_juego` y `renderizar_interfaz`: Representan visualmente el entorno del juego y los indicadores del jugador.
- `actualizar_stats_jugador`: Gestiona las barras de salud, energía, sed y hambre.

## OBSTÁCULOS PRESENTADOS Y SOLUCIONES:

El desarrollo del proyecto presentó varios retos técnicos y organizativos:

### 1. Problemas con las librerías SDL:

Durante la instalación y configuración de SDL en el entorno *msys*, enfrentamos problemas con archivos DLL faltantes. Esto se resolvió instalando las dependencias correctas y probando diversas configuraciones.

### 2. Organización de Archivos:

Al principio, los recursos gráficos estaban en la misma carpeta que el ejecutable, lo que dificultaba la organización. La solución fue usar rutas relativas con el prefijo `../recursos/`, permitiendo estructurar el proyecto de manera más lógica y ordenada.

### 3. Barras de Estadísticas:

Las barras de salud, hambre, sed y energía mostraban comportamientos erráticos, como valores negativos o caídas instantáneas. Para resolverlo, implementamos controles estrictos que garantizan que estos valores permanezcan dentro del rango esperado.

### 4. Cambios del clima:

La implementación de un sistema de clima dinámico (entre soleado y tormentoso) fue un proceso complejo que implicó dos días de pruebas y ajustes. Este sistema agregó un nivel adicional de realismo al juego.

### 5. Colaboración Asíncrona:

Dado que trabajamos de forma asíncrona, utilizamos un repositorio de GitHub para evitar conflictos en el código. Cada integrante actualizaba sus archivos locales antes de realizar cambios y subía sus avances al repositorio, avisando al equipo para garantizar la coherencia.

### 6. Detección y Corrección de Bugs:

La depuración fue una tarea intensiva. Un ejemplo destacado fue el diseño de las diferentes etapas del personaje, que cambian visualmente según el progreso del jugador (sin espada ni armadura, con espada y finalmente con ambos). La integración de estas texturas fue problemática al principio, pero logramos implementarlas con éxito mediante la edición en Canva y la remoción de fondos de las imágenes.

## LIBRERÍAS Y RECURSOS UTILIZADOS:

### Librerías:

- SDL2: Para gráficos y eventos.
- SDL\_image: Para manejar texturas e imágenes.

- SDL\_ttf: Para renderizar texto.
- SDL\_mixer: Para audio y efectos de sonido.

### **Recursos en línea:**

- Tutoriales de SDL: <https://lazyfoo.net/tutorials/SDL/>.
- Documentación oficial de SDL para manejo de texturas: [https://wiki.libsdl.org/SDL3/SDL\\_CreateTextureFromSurface](https://wiki.libsdl.org/SDL3/SDL_CreateTextureFromSurface).

### **Inteligencia Artificial:**

Consultamos a la IA para resolver dudas específicas sobre lenguaje C, como la implementación de punteros, validaciones en funciones complejas y la lógica detrás del sistema de combate.

### **CONCLUSIONES:**

El desarrollo del juego de la Isla de Supervivencia para el Laboratorio de C, representó un desafío apasionante que exigió no solo nuestras habilidades técnicas, sino también nuestra capacidad de colaborar eficazmente como equipo. Desde el primer momento, entendimos que este proyecto era la oportunidad de demostrar de lo que éramos capaces al crear un videojuego desde cero, enfrentando sus complejidades y sacando el máximo provecho de nuestras herramientas y conocimientos.

A lo largo del proceso, aprendimos la importancia de estructurar y modularizar el código. Este enfoque no solo facilitó el mantenimiento y la depuración, sino que también nos permitió integrar de forma eficiente los aportes de cada integrante del equipo.

GitHub se convirtió en nuestro aliado indispensable para el trabajo colaborativo. Al ser un proyecto con múltiples partes móviles, el control de versiones nos permitió sincronizar esfuerzos y evitar conflictos de código, lo que reforzó nuestra capacidad para organizarnos y trabajar en equipo, incluso de manera asíncrona.

Al mirar el resultado final, sentimos una mezcla de orgullo y satisfacción. Logramos implementar un juego funcional, con una experiencia gráfica y jugable que refleja el esfuerzo y la atención al detalle que dedicamos. Sin embargo, somos conscientes de que todo proyecto puede evolucionar y mejorar.

### **LINK EXPLICATIVO YOUTUBE:**

<https://www.youtube.com/watch?v=UPpgEXCMA9I>

### **LINK REPOSITORIO GITHUB FINAL:**

<https://github.com/EzePoss/Actualizador-Precios>

