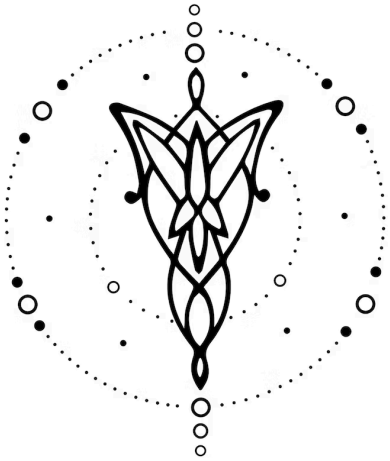

Entrega Final

Diseño e Implementación de un Lenguaje

1. Equipo

Nombre	Apellido	Legajo	E-mail
Emilio	Neme	62.601	emneme@itba.edu.ar
Gonzalo	Candisano	62.616	gcandisano@itba.edu.ar
Ben	Deyheralde	63.559	bdeyheralde@itba.edu.ar
Matías	Mutz	63.590	mmutz@itba.edu.ar

2. Repositorio

La solución y su documentación fueron versionadas en: **TLA_Messirve**. Como así futuras actualizaciones podrán verse allí.

3. Introducción

En 2022, se celebró la Copa Mundial de la FIFA, considerado el torneo de fútbol de mayor prestigio a nivel de selecciones, y un sueño anhelado por todos los jugadores profesionales. Cada edición de este mundial incluye su respectivo álbum de figuritas, una tradición que apasiona tanto a adultos como a niños. Sin embargo, en 2022, se produjo una notable escasez de estos álbumes, imposibilitando a muchos aficionados completar sus colecciones.

Con la próxima llegada de la Copa América, y anticipando la publicación de un nuevo álbum de figuritas, se ha desarrollado un lenguaje específico con el propósito de facilitar el diseño y la creación de estos álbumes, incluyendo tanto las figuritas clásicas como las especiales. Este enfoque busca evitar el desabastecimiento observado anteriormente, asegurando que todos los entusiastas puedan disfrutar y completar sus álbumes sin contratiempos.

4. Modelo Computacional

4.1. Dominio

El dominio del proyecto consiste en desarrollar un lenguaje que facilite la creación de figuritas personalizadas para cualquier torneo deseado. Este lenguaje está diseñado para permitir la generación de figuritas de diversos tipos, tales como jugadores, estadios, formaciones, escudos, camisetas, trofeos, pelotas utilizadas, y figuritas especiales.

El lenguaje debe proporcionar una sintaxis que sea intuitiva y sencilla tanto de entender como de programar. Cada figurita se especifica indicando su tipo y los parámetros correspondientes, siguiendo la documentación y respetando los parámetros requeridos.

Una vez definidos los parámetros, el lenguaje genera un archivo HTML que muestra la plantilla de la figurita de acuerdo con su tipo, integrando los campos proporcionados como parámetros. Las imágenes de las figuritas se deben suministrar mediante enlaces, los cuales se insertarán en el HTML.

La implementación efectiva de este lenguaje permitirá la creación rápida, eficiente y clara de figuritas, eliminando la necesidad de utilizar programas de diseño. Los usuarios solo deberán proporcionar los parámetros necesarios para la generación de las figuritas, simplificando así el proceso de diseño y facilitando la personalización de álbumes de figuritas para diferentes torneos.

4.2. Lenguaje

El lenguaje ofrece las siguientes funcionalidades:

- Se podrá crear la figura de un jugador mediante la declaración **player**.
    - Un jugador podrá tener atributos como **birthday, height, weight** y **team.**
- Se podrán crear torneos mediante la declaración **tournament**.
    - Dentro de un torneo, se pueden crear figuritas del trofeo, un estadio y una pelota mediante la declaración **trophy, stadium** y **ball**.
        - La declaración de **stadium** puede incluir el atributo **capacity.**
    - Los torneos pueden organizar equipos en grupo mediante la declaración **group.**
- Los equipos se definen mediante la declaración **team** que puede estar anidada a un **group.**
    - Un team podrá tener figuritas de jugadores, escudo (**badge**) y plantel (**homekit)**
- Todas las figuritas mencionadas anteriormente pueden tener el atributo **photo**

5. Implementación

5.1. Frontend

El desarrollo del frontend se centró en la creación de un lenguaje de dominio específico para la generación de figuritas de torneos, que permite a los usuarios definir figuritas personalizadas de jugadores, equipos, estadios y otros elementos.

La implementación incluyó algunos componentes clave. Primero que nada un lenguaje de entrada específico que permite definir figuritas de diferentes tipos mediante una sintaxis simple e intuitiva. Luego, para el análisis del lenguaje de entrada, se utilizaron **Flex** (Fast Lexical Analyzer generator) y **Bison** (GNU Parser Generator).

Flex se encargó de la tokenización del código de entrada, identificando los diferentes elementos del lenguaje. En el archivo “flex.Patterns.l” definimos los patrones que se iban a reconocer como tokens (player, torunament, stadium, entre otros). Además de los tokens, se definieron patrones como por ejemplo “digit” o “string” para luego asignarlo como atributo a un token.

Se utilizaron tres contextos, además del INITIAL:

- STRING: se utiliza para todo lo que vaya entre “:” y “;”.
- NAME: se utiliza para reconocer el nombre del tipo de dato. Por ejemplo, si el tipo de dato es “player”, lo que viene después de “player” es el nombre asignado a ese tipo de dato.
- MULTILINE_COMMENT: se utiliza para comentarios multilinea de forma “`/*  */`”.

Posteriormente, Bison se utilizó para el análisis sintáctico, construyendo el árbol de sintaxis abstracta (AST) que representa la estructura del código de entrada.

Para esta etapa se implementaron una serie de test para poder comprobar que el código funcione correctamente.

5.2. Backend

La generación del programa comienza con la función principal generate. Esta función abre un archivo de salida llamado *Output.html* y llama a una serie de funciones para construir el contenido del archivo HTML. La función _generateProgram se encarga de llamar a la función que va a empezar a crear el cuerpo del archivo de salida.

La función _generateElements es responsable de la generación de los cuerpos de elementos. Dependiendo del tipo de elemento (único o múltiple), se llama a la función _generateElement para cada elemento específico. Los elementos pueden ser de diferentes tipos, como trofeos, equipos, jugadores, estadios, escudos, pelotas, elementos especiales o torneos.

Para elementos que no son torneos ni equipos, se crea directamente la figurita correspondiente llamando a una función específica. Estas funciones generan el HTML correspondiente a cada elemento, incluyendo imágenes y detalles relevantes.

Cuando el elemento es un equipo, primero se verifica si el equipo ya ha sido creado previamente mediante una estructura llamada teamMap. Si el equipo no ha sido creado, se agrega al mapa y se procede a generar su HTML. La generación del equipo incluye la creación de sub-elementos como jugadores y otros detalles del equipo.

Para elementos que son torneos, se genera el HTML correspondiente incluyendo detalles adicionales como equipos participantes, grupos y otros elementos del torneo.

Una vez generados todos los elementos y sus detalles, el archivo HTML se cierra y se guarda, completando así la generación del output.

5.3. Adicionales

Como funcionalidad adicional, el equipo ha desarrollado una extensión para Visual Studio Code que mejora la experiencia de usuario al trabajar con el lenguaje creado para la generación de figuritas. Esta extensión proporciona soporte para resaltar la sintaxis del lenguaje específico, haciendo que el código sea más legible y visualmente organizado. Al escribir el código de entrada en Visual Studio Code, la extensión asegura que los diferentes elementos y estructuras se destaquen con colores apropiados, facilitando así la identificación de tipos de figuritas, atributos y otros componentes del lenguaje.

Esta característica adicional no solo mejora la estética del código, sino que también contribuye a reducir errores, mejorar la productividad y hacer el proceso de desarrollo más intuitivo. La integración de esta extensión en el flujo de trabajo del equipo permitió una programación más eficiente y un mejor manejo de la complejidad del lenguaje.

5.4. Dificultades encontradas

Durante el desarrollo del backend del proyecto, el equipo enfrentó varias dificultades. Una de las más significativas fue un error relacionado con la colocación de imágenes. Este problema

generó retrasos de varios días hasta que se descubrió que el error provenía de la omisión de una línea de código en el frontend. Identificar que el origen del problema estaba en una parte del sistema ya desarrollada previamente resultó ser un desafío considerable.

Otra dificultad importante fue la generación de la salida en formato HTML. Inicialmente, el equipo encontró complicado el proceso de crear el archivo de salida, especialmente debido a la necesidad de mantener la correcta indentación y estructuración del código HTML, que se formaba a partir de funciones, algunas de las cuales eran recursivas. La adaptación a este proceso requirió tiempo y esfuerzo.

Además, la idea de reutilizar equipos en el código, planteada originalmente como una tarea sencilla, demostró ser más compleja en la práctica. Para implementar esta funcionalidad, fue necesario diseñar una estructura de datos específica para los equipos, permitiendo que cada equipo se almacenara a medida que se definía. Cuando se encuentra una referencia a un equipo previamente definido, por ejemplo, `team Brasil`, el sistema busca en la estructura de datos y genera en la salida HTML los datos correspondientes de manera adecuada.

6. Futuras Extensiones

Durante el desarrollo del proyecto, el equipo se sintió motivado por la idea de que este trabajo tiene un potencial significativo si se considera la creación de una aplicación que permita a los usuarios elaborar álbumes con sus jugadores y equipos preferidos. Se reconoce que esta es una funcionalidad que no está ampliamente disponible o popularizada en el mercado actual. Por lo tanto, resulta interesante contemplar el lanzamiento de la misma en un futuro que ofrezca esta capacidad, incorporando las mejoras pertinentes.

Además, con más tiempo y recursos, se aspira a desarrollar una aplicación con una interfaz de usuario intuitiva y atractiva, que facilite la experiencia de creación de álbumes. El objetivo es proporcionar a los usuarios un entorno cómodo y una variedad de herramientas que les permitan personalizar sus álbumes según sus preferencias, asegurando que puedan alcanzar sus objetivos de forma más sencilla y agradable.

En cuanto a futuras funcionalidades, se plantea la posibilidad de expandir la creación de álbumes de figuritas a otras temáticas, más allá del fútbol, permitiendo así una mayor diversidad y personalización para los usuarios.

Se contempla también la incorporación de varias mejoras que amplíen la versatilidad y el alcance de la aplicación. Una de las principales mejoras previstas es la implementación de opciones de configuración de idioma. Actualmente, la aplicación opera exclusivamente con archivo de salida en español. Sin embargo, con miras a escalar la aplicación y llegar a un público más amplio, se planea ofrecer la posibilidad de seleccionar el idioma de preferencia del usuario, facilitando así su uso en diferentes regiones y contextos lingüísticos.

Además, se buscará añadir la capacidad de ingresar datos en diversas unidades de medida. Por ejemplo, se desea permitir que los usuarios introduzcan la altura en pies o en metros, y que la aplicación procese y muestre esta información en la unidad de medida elegida por el usuario. Esta funcionalidad mejorará la personalización y adaptación de la aplicación a las necesidades y preferencias específicas de cada usuario, haciéndola más accesible y flexible.

Estas mejoras están orientadas a aumentar la usabilidad y la accesibilidad de la aplicación, asegurando que pueda ser utilizada eficazmente en distintos contextos culturales y geográficos, y que ofrezca una experiencia más personalizada y adaptada a las preferencias de los usuarios.

7. Conclusiones

El equipo ha disfrutado enormemente trabajar en este proyecto, ya que ha permitido el aprendizaje de herramientas desconocidas hasta el momento. La experiencia resultó no solo educativa sino también divertida, especialmente al desarrollar una idea que, en un principio, se consideró tal vez como un chiste, pero que, al final, reveló su complejidad y llevó a una conclusión exitosa del trabajo.

Un aspecto que facilitó este logro fue la sólida base proporcionada por Agustín, lo que permitió al equipo concentrarse en alcanzar las funcionalidades propuestas desde el inicio del proyecto, asegurando que estas se implementarán sin problemas. La familiaridad y comodidad con el uso de Flex y Bison también fueron notables, dado que su similitud con los temas

abordados durante la cursada contribuyó a una adaptación más fluida, convirtiéndolas en herramientas valiosas para proyectos futuros.

Además, el proceso de probar los códigos en desarrollo mediante tests fue particularmente gratificante. La capacidad de evaluar las salidas en HTML y detectar posibles problemas para corregirlos en el momento facilitó la continuidad y mejora del trabajo, consolidando la experiencia como una práctica enriquecedora y satisfactoria.

8. Referencias

Extensión Visual Studio Code - Messirve Language:
https://marketplace.visualstudio.com/items?itemName=Messirve.messirve-language

9. Bibliografía