



UTMACH

UNIDAD ACADÉMICA DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE APLICACIÓN WEB PARA GESTIONAR UN
CAMPEONATO DE FÚTBOL USANDO METODOLOGÍA SNAIL Y
LENGUAJE DE PROGRAMACIÓN PYTHON

CAMACHO OSMOLIK ALEJANDRO VLADISLAV
INGENIERO DE SISTEMAS

MACHALA
2019



UTMACH

UNIDAD ACADÉMICA DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE APLICACIÓN WEB PARA GESTIONAR UN
CAMPEONATO DE FÚTBOL USANDO METODOLOGÍA SNAIL Y
LENGUAJE DE PROGRAMACIÓN PYTHON

CAMACHO OSMOLIK ALEJANDRO VLADISLAV
INGENIERO DE SISTEMAS

MACHALA
2019



UTMACH

UNIDAD ACADÉMICA DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

DESARROLLO DE APLICACIÓN WEB PARA GESTIONAR UN CAMPEONATO DE
FÚTBOL USANDO METODOLOGÍA SNAIL Y LENGUAJE DE PROGRAMACIÓN
PYTHON

CAMACHO OSMOLIK ALEJANDRO VLADISLAV
INGENIERO DE SISTEMAS

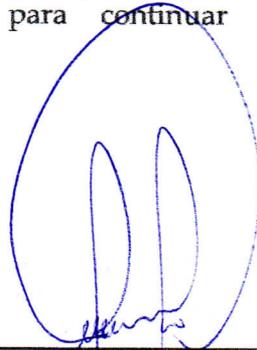
HONORES TAPIA JOOFRE ANTONIO

MACHALA, 04 DE FEBRERO DE 2019

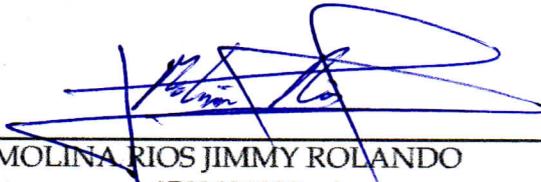
MACHALA
04 de febrero de 2019

Nota de aceptación:

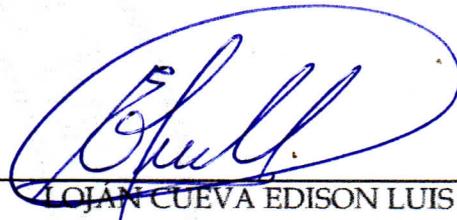
Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado DESARROLLO DE APLICACIÓN WEB PARA GESTIONAR UN CAMPEONATO DE FÚTBOL USANDO METODOLOGÍA SNAIL Y LENGUAJE DE PROGRAMACIÓN PYTHON, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.



HONORES TAPIA JOOFRE ANTONIO
0704811751
TUTOR - ESPECIALISTA 1



MOLINA RIOS JIMMY ROLANDO
0703691980
ESPECIALISTA 2



LOJAN CUEVA EDISON LUIS
0703249698
ESPECIALISTA 3

Fecha de impresión: lunes 04 de febrero de 2019 - 15:23

Urkund Analysis Result

Analysed Document: Examen Complexivo - Alejandro Camacho.docx (D46877252)
Submitted: 1/16/2019 6:51:00 PM
Submitted By: avcamacho_est@utmachala.edu.ec
Significance: 7 %

Sources included in the report:

GuinaVivianLuisFernando20181.pdf (D40261891)
TrabajoComplexivo-Castillo Jimmy .docx (D40217812)
INFORME-GÓMEZ.docx (D29675302)

Instances where selected sources appear:

5

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, CAMACHO OSMOLIK ALEJANDRO VLADISLAV, en calidad de autor del siguiente trabajo escrito titulado DESARROLLO DE APLICACIÓN WEB PARA GESTIONAR UN CAMPEONATO DE FÚTBOL USANDO METODOLOGÍA SNAIL Y LENGUAJE DE PROGRAMACIÓN PYTHON, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 04 de febrero de 2019

CAMACHO OSMOLIK ALEJANDRO VLADISLAV
0705788636

DEDICATORIA

A mi familia que ha estado apoyándome a conseguir todas las metas que me he propuesto tanto en mi preparación académica como en los demás aspectos de mi vida, educándome de la mejor manera para tener los valores que son necesarios para llegar a culminar mi educación formal.

Y a los docentes que me brindaron sus enseñanzas permitiéndome así obtener los conocimientos necesarios para finalmente cursar la universidad.

Sr. Camacho Osmolik Alejandro Vladislav

AGRADECIMIENTO

A mis padres que me brindaron el sustento económico necesario en mis estudios y me motivaron a seguir preparándome profesionalmente.

Agradezco también a los docentes que me prepararon académicamente en las diferentes materias.

Y finalmente a mis compañeros que fueron un apoyo en los diferentes trabajos grupales o me ayudaron a resolver dificultades que se presentaron a lo largo de la etapa académica.

Sr. Camacho Osmolik Alejandro Vladislav

RESUMEN

DESARROLLO DE APLICACIÓN WEB PARA GESTIONAR UN CAMPEONATO DE FÚTBOL USANDO METODOLOGÍA SNAIL Y LENGUAJE DE PROGRAMACIÓN PYTHON

En el presente trabajo se desarrolló una página web para gestionar un campeonato de fútbol el cual deberá poder agregar equipos, crear partidos teniendo en cuenta que los equipos jugaran todos contra todos, generar una tabla de posiciones dinámica que muestre los partidos ganados, partidos empatados y partidos perdidos, considerando los goles a favor y en contra obteniendo así su gol diferencia por cada equipo participante. Además, una vez terminada la primera fase del campeonato que se determina cuando todos los equipos hayan jugado contra todos se deberá pasar a la siguiente fase donde el primer equipo en la tabla de posiciones jugara contra el tercero y el segundo contra el cuarto determinando finalmente cuales equipos pasaran a disputar el tercer lugar y cuales pasaran a disputar la final. Como editor de código se utilizó Sublime Text con el cual se escribió el código para el lado del servidor que es Python con el Framework de desarrollo Django y del lado del cliente HTML5 para el maquetado de la aplicación web, para los estilos se usó CSS con el Framework Bootstrap y finalmente JQuery para brindar interacción al usuario. El almacenamiento de la información se realizó mediante PostgreSQL como sistema gestor de base de datos relacional y para cada fase de desarrollo de la aplicación web se utilizó como guía la metodología híbrida SNAIL.

Palabras Claves: Aplicación Web, Django, Framework, Python, Servidor Web, Metodología SNAIL

ABSTRACT

DEVELOPMENT OF A WEB APPLICATION TO MANAGE A SOCCER CHAMPIONSHIP USING SNAIL METHODOLOGY AND PYTHON PROGRAMMING LANGUAGE

In the present work a web page was developed to manage a football championship which should be able to add teams, create matches taking into account that the teams will play all against all, generate a dynamic stand that shows the matches won, matches tied and lost games, considering the goals for and against obtaining his goal difference for each participating team. In addition, once the first phase of the championship is finished, which is determined when all the teams have played against all, the next stage will be played where the first team in the standings will play against the third and the second against the fourth, finally determining which teams will go to dispute the third place and which will go to dispute the final. As a code editor we used Sublime Text with which we wrote the code for the server side that is Python with the Django Development Framework and the HTML5 client side for the layout of the web application, for the styles we used CSS with The Bootstrap Framework and finally JQuery to provide user interaction. The storage of the information was carried out using PostgreSQL as a relational database management system and for each development phase of the web application the SNAIL hybrid methodology was used as a guide.

Keywords: Web Application, Django, Framework, Python, Web Server, SNAIL Methodology

CONTENIDO

	pág.
DEDICATORIA	1
AGRADECIMIENTO	2
RESUMEN	3
ABSTRACT	4
CONTENIDO	5
LISTA DE TABLAS	7
LISTA DE ILUSTRACIONES	8
1. INTRODUCCIÓN	9
1.1 Marco Contextual	10
1.2 Problema	10
1.3 Objetivo General	10
2. DESARROLLO	11
2.1 Marco Teórico	11
2.1.1 <i>Aplicaciones web</i>	11
2.1.2 <i>Computación en la nube</i>	11
2.1.3 <i>Python</i>	11
2.1.4 <i>Framework Django</i>	11
2.1.5 <i>Arquitectura lógica: Modelo, Vista, Template</i>	12
2.1.6 <i>Reportes: xhtml2pdf</i>	12
2.1.7 <i>PostgreSQL</i>	12
2.1.8 <i>HTML5</i>	12
2.1.9 <i>JavaScript</i>	13
2.1.10 <i>Responsive Design</i>	13
2.1.11 <i>Protocolo de Comunicación HTTP</i>	13
2.1.12 <i>Servidor Web</i>	13

2.1.13	<i>Sistema operativo Linux</i>	13
2.1.14	<i>Metodología SNAIL</i>	14
2.1.15	<i>Fases de la metodología SNAIL</i>	14
2.2	Solución del Problema	14
2.2.1	<i>Materiales</i>	14
2.2.2	<i>Desarrollo de la metodología SNAIL</i>	14
2.2.2.1	<i>Fase 1: Requisitos</i>	14
2.2.2.2	<i>Fase 2: Planificación</i>	15
2.2.2.3	<i>Fase 3: Diseño</i>	15
2.2.2.4	<i>Fase 4: Programación</i>	15
2.2.2.5	<i>Fase 5: Pruebas</i>	16
2.2.2.6	<i>Fase 6: Clausura</i>	16
2.3	Resultados	16
3.	CONCLUSIONES	18
	BIBLIOGRAFÍA	19
	ANEXOS	21

LISTA DE TABLAS

	pág.
Tabla 1. Materiales utilizados	14
Tabla 2. Gestión de Usuario - RF001	21
Tabla 3. Gestión de Campeonato - RF002	21
Tabla 4. Gestión de Equipos - RF003	21
Tabla 5. Comenzar Campeonato - RF004	22
Tabla 6. Jugar Partido - RF005	22
Tabla 7. Gestionar tabla de Posiciones- RF006	23
Tabla 8. Visualizar Ganadores - RF007	23
Tabla 9. Generar Reportes - RF008	23
Tabla 10. Validaciones - RNF001	24
Tabla 11. Diseño Responsivo - RNF002	24
Tabla 12. Control de usuarios - RNF003	24

LISTA DE ILUSTRACIONES

	pág.
Ilustración 1. Modelo Entidad-Relación	25
Ilustración 2. Modelo Relacional	25
Ilustración 3. Prototipo de Login	26
Ilustración 4. Prototipo Campeonato	26
Ilustración 5. Prototipo Partidos	26
Ilustración 6. Inicio de Sesión	27
Ilustración 7. Registro de usuario	27
Ilustración 8. Ingresar Campeonato	28
Ilustración 9. Editar y eliminar campeonato	28
Ilustración 10. Crear equipo	29
Ilustración 11. Editar y eliminar equipo	29
Ilustración 12. Listado de partidos	30
Ilustración 13. Jugar partido	30
Ilustración 14. Tabla de posiciones	31
Ilustración 15. Ganadores no definidos	31
Ilustración 16. Ganadores campeonato	32

1. INTRODUCCIÓN

El protagonismo del internet desde hace muchos años ha de gran importancia para las empresas que han cambiado sus formas de manejar la información migrando a la computación en la nube la cual proporciona diferentes ventajas entre las cuales están el poder acceder a la información desde cualquier lugar del mundo y realizar procesos complejos sin la necesidad de que el usuario conozca la infraestructura involucrada detrás de las aplicaciones y servicios. [1]

Las herramientas tecnológicas son cada vez más sencillas de usar para que hasta los usuarios menos experimentados puedan pasar de un manejo manual de sus procesos a la computación en la nube que es esencial para mejorar los procesos de las empresas y ayudan de esta forma a la reducción de costos.

Para acceder a las aplicaciones web no es necesario de equipos con características elevadas, solo se necesita de un dispositivo con acceso a internet el cual realizara la petición a los servidores que son los encargados de realizar los procesos y enviar la información al usuario de manera rápida, segura y disponible en cualquier momento para los usuarios autorizados. [2]

En este proyecto se tiene como propósito crear una aplicación web para la gestión de un campeonato de fútbol que satisfaga todos los procesos necesarios y brinde al usuario una plataforma sencilla y eficiente.

Las principales tecnologías a utilizar en la aplicación web son Python con Django del lado del servidor, PostgreSQL para el almacenamiento de los datos y para el correcto desarrollo del proyecto se utilizó la metodología híbrida SNAIL.

El presente informe de titulación se ha estructurado de la siguiente manera:

Capítulo 1: Contiene información generalizada del proyecto, el problema a resolver y el objetivo general.

Capítulo 2: Marco teórico, solución al problema y resultados.

Capítulo 3: Conclusiones, bibliografía y anexos.

1.1 Marco Contextual

Los campeonatos de futbol tienen diferentes etapas que las personas encargadas de organizarlo deben tomar en cuenta durante todo el proceso del mismo siendo necesario una aplicación web que maneje los diferentes procesos como son el gestionar los equipos, partidos, resultados y el manejo de las puntuaciones necesarias para determinar los ganadores.

En el presente caso de estudio se desarrollará un campeonato en el cual todos los equipos jugaran contra todos en la primera etapa, en la segunda fase se jugará entre los primeros cuatro equipos en la tabla de posiciones con lo cual se determinarán mediante los resultados obtenidos los equipos que pasen a disputar el tercer lugar y la final.

1.2 Problema

Las personas encargadas de organizar campeonatos de futbol requieren gestionar de manera automatizada los procesos involucrados ya que por lo general son manejados manualmente registrando los equipos participantes, creando partidos y llevando una bitácora de los resultados por cada encuentro, pero esto puede generar errores en las puntuaciones de cada equipo, por lo cual se necesita de una aplicación web que garantice la exactitud de las puntuaciones que son actualizadas por cada partido jugado.

1.3 Objetivo General

Desarrollar una aplicación web para automatizar los procesos de un campeonato de futbol mediante el uso de Python y el framework Django

2. DESARROLLO

2.1 Marco Teórico

2.1.1 *Aplicaciones web*

En la actualidad, las aplicaciones web han surgido exponencialmente debido a las diferentes ventajas que ofrecen y a la dependencia del internet para el desarrollo de todo tipo de procesos dejando de lado a las convencionales aplicaciones de escritorio que se utilizaban como medio de difusión de información y otros servicios. [3]

2.1.2 *Computación en la nube*

La computación en la nube es un modelo innovador que provee servicios de forma centralizada usando Internet como medio de distribución. La propia arquitectura permite que el usuario pueda acceder a la capacidad de servidores con características elevadas que realizan cálculos complejos. [4]

2.1.3 *Python*

Python es un lenguaje de programación que cuenta con facilidades para la programación orientada a objetos, imperativa y funcional, lo cual lo vuelve un lenguaje multiparadigma. Es un lenguaje de alto nivel ya que tiene implícitas diferentes estructuras de datos que permiten realizar operaciones complejas con pocas líneas de código. [5]

2.1.4 *Framework Django*

Django es un framework de código abierto para desarrollar aplicaciones web escrito en Python el cual implementa el modelo, vista, template y provee al desarrollador de un conjunto de componentes que le permiten crear sitios de manera más rápida y eficiente. [6]

2.1.5 *Arquitectura lógica: Modelo, Vista, Template*

Las aplicaciones interactúan con los usuarios y presentan datos. La separación de interacción, presentación y datos es lo que define el Modelo Vista Template (MVT) que además tiene diferentes características que ayudan a crear mejores aplicaciones como son modularidad, seguridad en los datos, escalabilidad y rendimiento. [7]

2.1.6 *Reportes: xhtml2pdf*

Es un convertidor de html2pdf que está escrito en Python y utiliza el kit de herramientas de ReportLab, HTML5lib y pyPdf. Es compatible con HTML 5 y CSS 2.1 y algunos CSS 3. Esta librería permite a los usuarios generar un documento PDF de sus páginas realizadas con HTML Y CSS sin la necesidad de utilizar herramientas complejas o aprender nuevas tecnologías. [8]

2.1.7 *PostgreSQL*

Es un sistema de gestión de bases de datos relación, de código fuente libre. Es el gestor de bases de datos de más potente en la actualidad. Utiliza un modelo cliente-servidor utilizando multiprocesos para asegurar la estabilidad del sistema garantizando que el sistema siga funcionando en caso de haber un problema en uno de sus procesos. [9]

2.1.8 *HTML5*

Es un lenguaje basado en la web que casi todos los navegadores de Internet pueden usar [10]. HTML5 crea la estructura de los contenidos de los sitios web haciendo uso de etiquetas por cual es llamado un lenguaje de marcado que da significado a la información en los sitios web, estas etiquetas además ayudan a mejorar el posicionamiento en los buscadores. [11]

2.1.9 *JavaScript*

JavaScript es el lenguaje de programación del lado del navegador asíncrono que se ha vuelto mucho más poderoso y actualmente también puede ejecutarse del lado del servidor. Los motores de JavaScript eran intérpretes lentos, pero hoy en día son mucho más rápidos y efectivos mediante el uso de técnicas de compilación en tiempo real como el motor de código abierto V8.[12]

2.1.10 *Responsive Design*

El diseño responsivo para la web usa diferentes técnicas de diseño que permiten a los sitios web adaptarse a los diferentes formatos de dispositivos brindando una mejor experiencia al usuario, permitiendo la navegación en computadores, dispositivos móviles, televisores y demás medios existentes para poder acceder a internet. [13]

2.1.11 *Protocolo de Comunicación HTTP*

HTTP (Protocolo de transporte de hipertexto) se utiliza para construir servicios web [14]. Es un protocolo típico para solicitar y transmitir información, que controla la transferencia de datos entre el servidor y el cliente (como un navegador web). [15]

2.1.12 *Servidor Web*

Un servidor web es un software que está diseñado para recibir solicitudes por los usuarios por medio del protocolo HTTP, la respuesta que envía hacia el cliente es una transferencia de archivos que son generalmente renderizados por el navegador. [16]

2.1.13 *Sistema operativo Linux*

Linux es un sistema operativo libre que ofrece listas de aplicaciones agrupadas en diferentes categorías logrando que los usuarios se adaptan fácilmente al nuevo entorno. [17] Las tecnologías basadas en Linux se

han vuelto muy populares entre los desarrolladores de software y los administradores de sistemas.[18]

2.1.14 Metodología SNAIL

SNAIL (Software Nativo de Arquitectura Iterativa Lógica), es una metodología híbrida de desarrollo de aplicaciones web, que se basa en la simplicidad, comunicación y planificación del código desarrollado, ya que su modelo es en espiral toma la forma de un caracol de donde su nombre nace. [19]

2.1.15 Fases de la metodología SNAIL

La metodología presenta las siguientes fases: Requisitos, Panificación, Diseño, Programación, Pruebas, Clausura e Inboud Marketing que es una fase opcional en la metodología. [19]

2.2 Solución del Problema

2.2.1 Materiales

Los materiales utilizados para el caso de estudio son los siguientes:

Tabla 1. Materiales utilizados

Materiales / Recursos	
Lenguaje de Programación	Python 2.7.13
Framework	Django 1.11.11
Gestor de base de datos	PostgreSQL 9.4.4
Metodología	SNAIL
Servidor	S.O: Ubuntu; Servidor web: Apache

Fuente: Elaboración propia

2.2.2 Desarrollo de la metodología SNAIL

2.2.2.1 Fase 1: Requisitos

En esta fase se define que debe realizar la aplicación web mediante el desarrollo del software para cubrir las necesidades

del cliente. Un requisito debe solucionar cierto problema real del mundo real y deben ser descritos de manera clara ya que un fallo en la correcta recolección de los mismos afectara a las demás etapas del proyecto.

Para documentar todos los requisitos funcionales y no funcionales que deben ser cubiertos, se realizara una plantilla por cada requisito (Ver Anexo A).

2.2.2.2 *Fase 2: Planificación*

En esta etapa se brinda un framework que permite al gestor hacer estimaciones razonables de recursos, entregables, costos y planificación temporal. Estas estimaciones se realizan dentro de un tiempo limitado al comenzar el proyecto de software y tendrán que ser revisados constantemente a medida que el proyecto se va desarrollando. También las estimaciones deberían definir los dos escenarios tanto el optimista como el pesimista, de modo que los resultados del proyecto pueden limitarse.

2.2.2.3 *Fase 3: Diseño*

El diseño de la aplicación web debe basarse en diseño simples y puntuales que logren satisfacer las necesidades del usuario.

Tomando en cuenta los requisitos que se deben cumplir se pasó a realizar el diseño de la base de datos mediante el modelo entidad – relación (Ver Anexo B) y posteriormente el modelo relacional (Ver Anexo C)

También se realizaron prototipos de la interfaz gráfica (Ver anexo D) con lo cual se pretende que el cliente tenga una idea más clara de cómo se verá la aplicación web cuando se termine la fase de programación.

2.2.2.4 *Fase 4: Programación*

En esta fase se empieza a escribir el código fuente que deberá cumplir los requerimientos del usuario recolectados en la primera

fase de metodología teniendo en cuenta las buenas practicas al momento de programar creando un código legible, consistente y fácil de escalar. Los errores que vayan surgiendo del proceso de programación deben ir siendo corregidos y realizar nuevas pruebas para comprobar la correcta solución del error.

El cliente debe estar presente en la etapa de programación ya que los requerimientos son de alto nivel y no suelen proporcionar detalles específicos.

2.2.2.5 *Fase 5: Pruebas*

Los diferentes módulos creados mediante las iteraciones realizadas en la metodología se juntan para comprobar que cumplan con todos los requisitos solicitados por el cliente.

Las pruebas pueden ser de diferentes tipos como unitarias, de integración, de sistema y pruebas de aceptación con lo cual se asegura que el sistema se encuentre funcionando correctamente.

2.2.2.6 *Fase 6: Clausura*

Esta fase es la más corta de la metodología, pero no menos importante ya que se realiza el cierre del proyecto y se analiza los errores cometidos para el mejoramiento continuo a futuro.

También debe definir el estado de los entregables al finalizar el proyecto los cuales pueden ser completado, abandonado, pospuesto, cerrado. Al final se debe declara el éxito o fracaso del proyecto por las variables de cumplimiento de los requisitos, tiempo y presupuesto.

2.3 Resultados

Después de la aplicación de la metodología se obtuvieron los siguientes resultados en el desarrollo de la aplicación web para gestionar un campeonato de futbol con las siguientes funcionalidades:

Gestión de usuarios: La aplicación web cuenta con un inicio de sesión (Ver anexo E) en el cual se debe ingresar el nombre de usuario y contraseña, para crear un nuevo usuario (Ver Anexo F) se debe ingresar el nombre de usuario y una contraseña dos veces.

Gestión de Campeonato: Para gestionar un campeonato se debe crear mediante la siguiente interfaz (Ver anexo G) ingresando un nombre y la url del logo. Una vez creado se puede editar presionando encima del logo y eliminar (Ver Anexo H)

Gestión de Equipos: Para gestionar un equipo se debe crear mediante la siguiente interfaz (Ver anexo I) ingresando un nombre y la url del logo. Una vez creado se puede editar presionando encima del logo y eliminar (Ver Anexo J)

Gestión de Partidos: Para gestionar los partidos el campeonato debe haber comenzado y se podrá ver una lista de partidos (Ver Anexo K) los cuales se podrán jugar ingresando el número de goles por cada partido (Ver Anexo L).

Tabla de Posiciones: La tabla de posiciones muestra los equipos participantes del campeonato ordenados dependiendo la puntuación que han ido obteniendo en los partidos (Ver Anexo M)

Ganadores: Los ganadores en caso de no estar definidos tendrán imágenes genéricas que identifiquen la posición que tendrán dependiendo el puesto que ha quedado el equipo (Ver Anexo N) y una vez definidos cambiarán las imágenes por el logo y nombre del respectivo equipo (Ver Anexo Ñ)

Reportes: Finalmente la aplicación genera un reporte de los equipos que pasaron a la segunda fase del campeonato y los equipos ganadores.

3. CONCLUSIONES

El desarrollo de la aplicación web para el gestionamiento de un campeonato de futbol se realizó exitosamente mediante la aplicación de la metodología SNAIL para la correcta recolección de los requisitos y demás fases de desarrollo del proyecto.

Para la fase de programación se utilizó el framework Django que permite un desarrollo rápido, seguro y escalable que son características de vital importancia para las aplicaciones web modernas, además de cumplir con la automatización de todos los procesos involucrados en un campeonato de futbol con una interfaz sencilla y amigable que permita al usuario su fácil utilización.

Al ser una aplicación web brinda a los usuarios la posibilidad de gestionar su campeonato desde cualquier dispositivo inteligente con acceso a internet y reduciendo considerablemente la posibilidad de pérdida de información al estar los datos almacenados en un servidor web.

BIBLIOGRAFÍA

- [1] R. Miralles, "Cloud computing y protección de datos," *IDP. Rev. Internet, Derecho y Política*, p. 11, 2010.
- [2] N. Lisbeth and H. Quintero, "Computación en la nube," *Rev. Mundo FESC*, pp. 46–51, 2014.
- [3] J. Molina, M. Zea, M. Contenido, and F. García, "Comparación de metodologías en aplicaciones web," *3C Technol. glosas innovación Apl. a la pyme*, vol. 7, pp. 1–19, 2018.
- [4] J. Calvo, J. Gracia, and E. Bayo, "Aplicación web para el análisis y diseño de estructuras," *Inf. la Construcción*, vol. 66, pp. 1–8, 2014.
- [5] O. L. Moya, "El lenguaje de programación Python," *Cent. Inf. y Gestión Tecnológica Santiago Cuba Cuba*, vol. 20, pp. 1–13, 2014.
- [6] J. Molina, N. Mora, M. Zea, and E. Loaiza, "Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python," *Rev. Latinoam. Ing. Softw.*, vol. 4, no. 4, pp. 201–207, 2016.
- [7] M. C. Enache, "Web Application Frameworks," *Dunarea Jos Univ. Galati*, pp. 82–86, 2015.
- [8] C. Glass, "xhtml2pdf 0.0.2," *Python Software Foundation*, 2011. [Online]. Available: <https://pypi.org/project/xhtml2pdf/0.0.2/>.
- [9] M. P. Zea Ordoñez, J. R. Molina Ríos, and F. F. Redrován Castillo, *Administración de bases de datos con PostgreSQL*. 2017.
- [10] M. Muñoz and M. Hernández, "Software development to increase coping skills in adolescents," *J. Behav. Heal. Soc. Iss ues*, vol. 9, pp. 7–12, 2017.
- [11] M. Guerra, C. González, and J. Pedro, "Diseño web adaptativo para la plataforma educativa ZERA," *Rev. Cuba. Ciencias Informáticas*, vol. 10, pp. 100–115, 2016.
- [12] P.-G. Sprimont, D. Ricci, and L. Nicastro, "New web technologies for astronomy," *Rev. Mex. Astron. y Astrofis.*, vol. 45, pp. 75–78, 2014.
- [13] D. Gonzales and M. C. Marcos, "Responsive web design : diseño multidispositivo para mejorar la experiencia de usuario," *Textos Univ. Bibliotecon.*, pp. 1–10, 2015.

- [14] T. Yuan *et al.*, “Formalization and Verification of REST on HTTP Using CSP,” *Electron. Notes Theor. Comput. Sci.*, vol. 309, pp. 75–93, 2014.
- [15] M. Sysel and O. Dolež, “An Educational HTTP Proxy Server,” *Procedia Eng.*, vol. 69, pp. 128–132, 2014.
- [16] F. Ríos, F. Polanco, and V. Moreno, “Servidor web empotrado en un FPGA para configurar un Controlador Maestro del Sistema Inteligente de Tráfico Cubano FPGA,” *Rev. Cuba. Ciencias Informáticas*, vol. 11, pp. 16–28, 2017.
- [17] P. Fernández *et al.*, “Equivalencia de aplicaciones para la Plataforma GNU/LINUX,” *Panor. Cuba y Salud*, vol. 6, pp. 39–41, 2011.
- [18] J. Gomes *et al.*, “Enabling rootless Linux Containers in multi-user environments : The udocker tool,” *Comput. Phys. Commun.*, vol. 232, pp. 84–97, 2018.
- [19] J. Molina, F. Redrovan, and N. Loja, *SNAIL, Una metodología híbrida para el desarrollo de aplicaciones web*, no. June. 2018.

ANEXOS

Anexo A: Requisitos funcionales y no funcionales

Tabla 2. Gestión de Usuario - RF001

Identificador del requisito	RF - 001
Nombre del requisito	Gestión de Usuario
Tipo	Funcional
Descripción	Se podrán crear usuarios para la gestión de campeonatos
Precondición	Ninguna
Actores	Usuario
Prioridad	Alta
Condición de Éxito	Se pudo crear un usuario exitosamente.
Condición de Fracaso	Los campos son incorrectos al gestionar el usuario

Fuente: Elaboración propia

Tabla 3. Gestión de Campeonato - RF002

Identificador del requisito	RF - 002
Nombre del requisito	Gestión de Campeonato
Tipo	Funcional
Descripción	Se podrán gestionar un campeonato de futbol
Precondición	Ninguna
Actores	Usuario
Prioridad	Alta
Condición de Éxito	Se pudo crear el campeonato de futbol y gestionarlo.
Condición de Fracaso	Los campos son incorrectos al gestionar el campeonato

Fuente: Elaboración propia

Tabla 4. Gestión de Equipos - RF003

Identificador del requisito	RF - 003
Nombre del requisito	Gestión de Equipos
Tipo	Funcional

Descripción	Se podrán gestionar un equipo de futbol.
Precondición	Ninguna
Actores	Usuario
Prioridad	Alta
Condición de Éxito	Se pudo crear el equipo y gestionarlo.
Condición de Fracaso	Los campos son incorrectos al gestionar el equipo

Fuente: Elaboración propia

Tabla 5. Comenzar Campeonato - RF004

Identificador del requisito	RF - 004
Nombre del requisito	Comenzar campeonato
Tipo	Funcional
Descripción	Se podrá comenzar el campeonato de futbol y automáticamente se crearan los partidos a disputar
Precondición	Deben existir al menos 8 equipos agregados
Actores	Usuario
Prioridad	Alta
Condición de Éxito	Se pudo comenzar el campeonato y se crearon los partidos.
Condición de Fracaso	No se puede iniciar el campeonato

Fuente: Elaboración propia

Tabla 6. Jugar Partido - RF005

Identificador del requisito	RF - 005
Nombre del requisito	Jugar Partido
Tipo	Funcional
Descripción	Se podrá jugar un partido asignando los goles para cada equipo participante.
Precondición	El partido debe haber sido creado
Actores	Usuario
Prioridad	Alta
Condición de Éxito	El partido se jugó y se asignan los goles para cada partido
Condición de Fracaso	Los valores de los goles no son de tipo numérico y el partido no se puede guardar.

Fuente: Elaboración propia

Tabla 7. Gestionar tabla de Posiciones- RF006

Identificador del requisito	RF - 006
Nombre del requisito	Gestionar tabla de Posiciones
Tipo	Funcional
Descripción	Por cada partido jugado la tabla de posiciones debe actualizarse situando al equipo con más puntos en primer lugar
Precondición	Deben existir partidos jugados
Actores	Sistema
Prioridad	Alta
Condición de Éxito	Se puede visualizar los equipos en la tabla de posiciones con sus respectivos puntos
Condición de Fracaso	No se muestran valores en la tabla de posiciones

Fuente: Elaboración propia

Tabla 8. Visualizar Ganadores - RF007

Identificador del requisito	RF - 007
Nombre del requisito	Visualizar Ganadores
Tipo	Funcional
Descripción	Al terminar el partido de tercer lugar se podrá visualizar el equipo que quedo en tercer lugar y al jugar la final se mostrara el campeón y vice campeón
Precondición	Se jugó el partido de final y tercer lugar
Actores	Sistema
Prioridad	Media
Condición de Éxito	Se puede los equipos ganadores del tercer lugar, segundo lugar y primer lugar del campeonato
Condición de Fracaso	No se muestran los equipos ganadores.

Fuente: Elaboración propia

Tabla 9. Generar Reportes - RF008

Identificador del requisito	RF - 008
Nombre del requisito	Generar Reportes
Tipo	Funcional
Descripción	Al terminar partido de la final se podrá generar un reporte de los equipos que clasificaron a la segunda fase y los ganadores del campeonato

Precondición	Se jugó el partido de final
Actores	Sistema
Prioridad	Media
Condición de Éxito	Se puede genera un reporte con los equipos clasificados a segunda fase y ganadores
Condición de Fracaso	No se muestra el reporte.

Fuente: Elaboración propia

Tabla 10. Validaciones - RNF001

Identificador del requisito	RNF - 001
Nombre del requisito	Validaciones
Tipo	No Funcional
Descripción	La aplicación web debe validar el correcto ingreso de datos.
Prioridad	Media

Fuente: Elaboración propia

Tabla 11. Diseño Responsivo - RNF002

Identificador del requisito	RNF - 002
Nombre del requisito	Diseño Responsivo
Tipo	No Funcional
Descripción	La aplicación web debe adaptarse a diferentes formatos de pantalla.
Prioridad	Media

Fuente: Elaboración propia

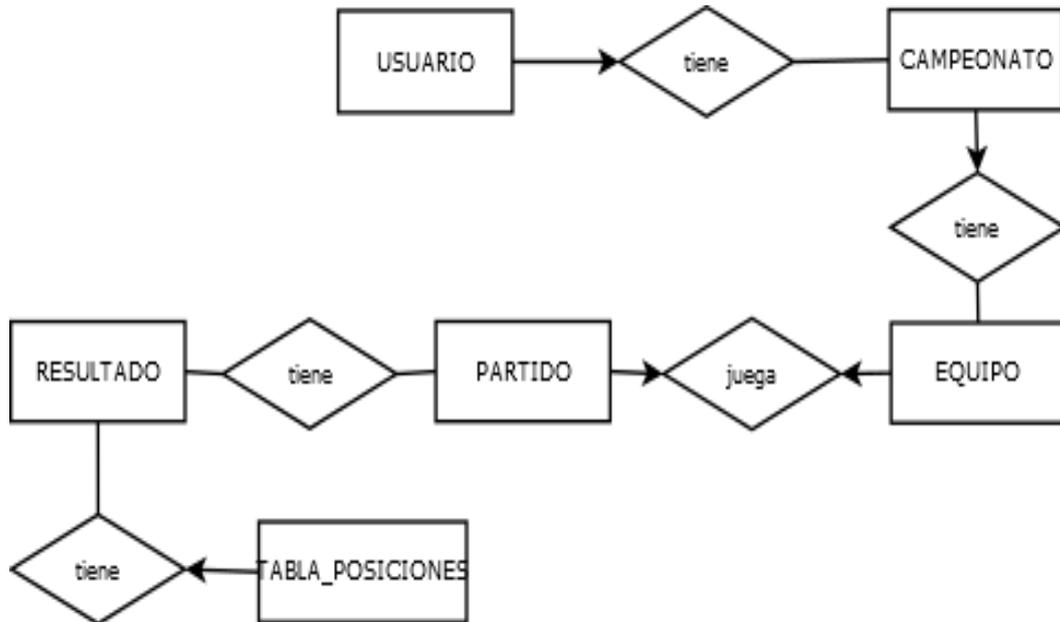
Tabla 12. Control de usuarios - RNF003

Identificador del requisito	RNF - 003
Nombre del requisito	Control de usuarios
Tipo	No Funcional
Descripción	La aplicación web permitirá acceder solamente a los usuarios registrados a las funcionalidades.
Prioridad	Alta

Fuente: Elaboración propia

Anexo B: Modelo Entidad – Relación

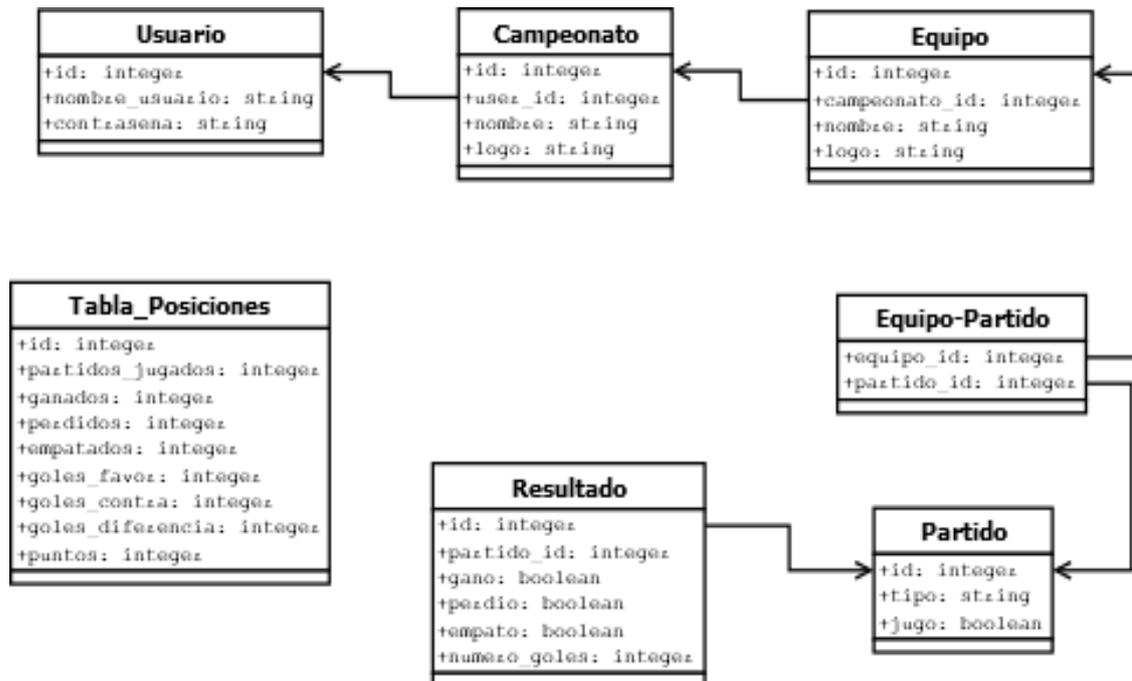
Ilustración 1. Modelo Entidad-Relación



Fuente: Elaboración propia

Anexo C: Modelo Relacional

Ilustración 2. Modelo Relacional



Fuente: Elaboración propia

Anexo D: Diseño de Interfaz

Ilustración 3. Prototipo de Login



Logo placeholder (X) labeled LOGO.

La mejor forma de gestionar tu campeonato!

Inicio Sesión Crear Cuenta

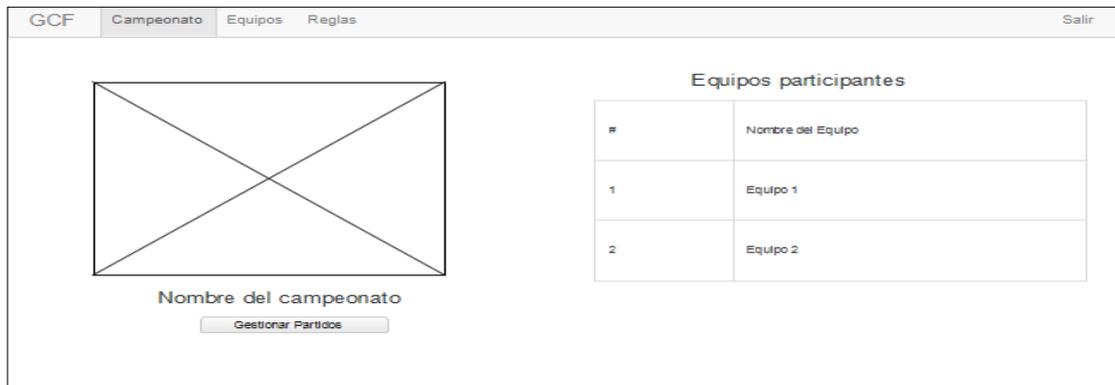
Ingresar usuario

Ingresar Contraseña

INICIAR SESIÓN

Fuente: Elaboración propia

Ilustración 4. Prototipo Campeonato



GCF Campeonato Equipos Reglas Salir

Nombre del campeonato

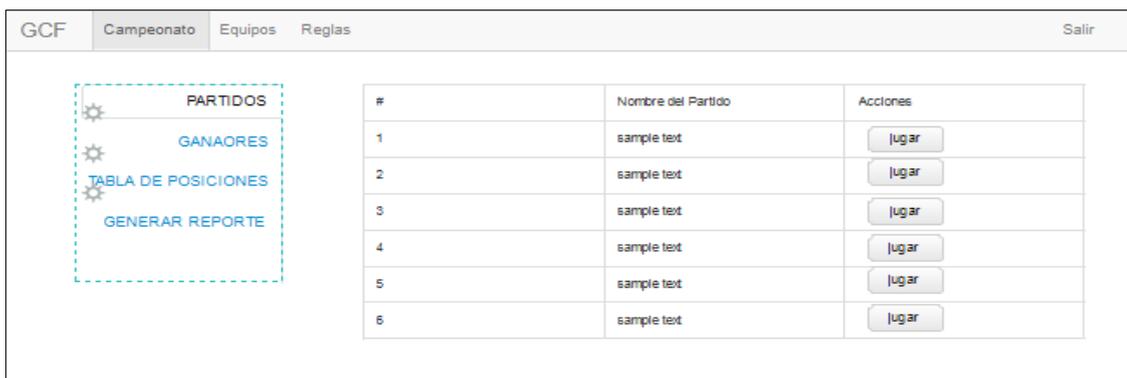
Gestionar Partidos

Equipos participantes

#	Nombre del Equipo
1	Equipo 1
2	Equipo 2

Fuente: Elaboración propio

Ilustración 5. Prototipo Partidos



GCF Campeonato Equipos Reglas Salir

PARTIDOS

GANADORES

TABLA DE POSICIONES

GENERAR REPORTE

#	Nombre del Partido	Acciones
1	sample text	Jugar
2	sample text	Jugar
3	sample text	Jugar
4	sample text	Jugar
5	sample text	Jugar
6	sample text	Jugar

Fuente: Elaboración propia

Anexo E: Inicio de Sesión

Ilustración 6. Inicio de Sesión



Fuente: Elaboración propia

Anexo F: Registrar usuario

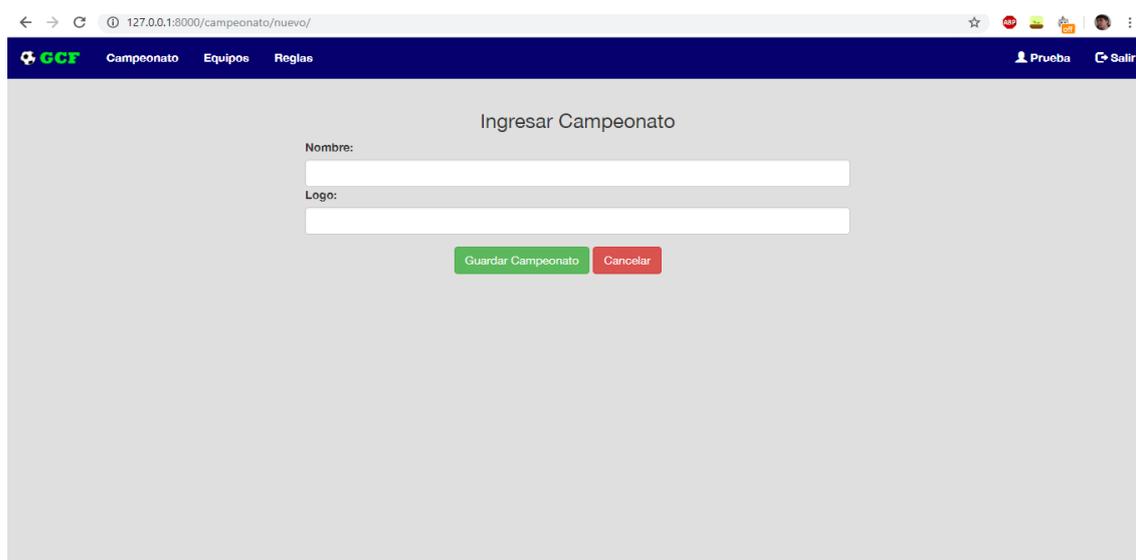
Ilustración 7. Registro de usuario



Fuente: Elaboración propia

Anexo G: Crear Campeonato

Ilustración 8. Ingresar Campeonato

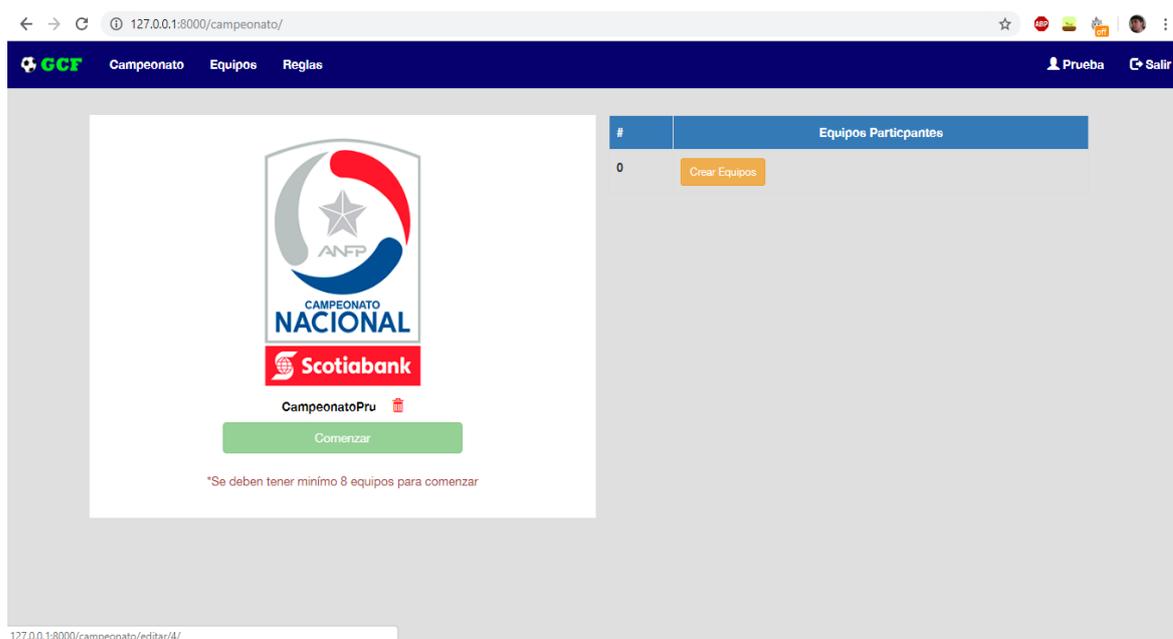


The screenshot shows a web browser window with the URL `127.0.0.1:8000/campeonato/nuevo/`. The page header includes the GCF logo and navigation links for 'Campeonato', 'Equipos', and 'Reglas'. On the right, there are links for 'Prueba' and 'Salir'. The main content area is titled 'Ingresar Campeonato' and contains two input fields: 'Nombre:' and 'Logo:'. Below these fields are two buttons: a green 'Guardar Campeonato' button and a red 'Cancelar' button.

Fuente: Elaboración propia

Anexo H: Editar y eliminar campeonato

Ilustración 9. Editar y eliminar campeonato

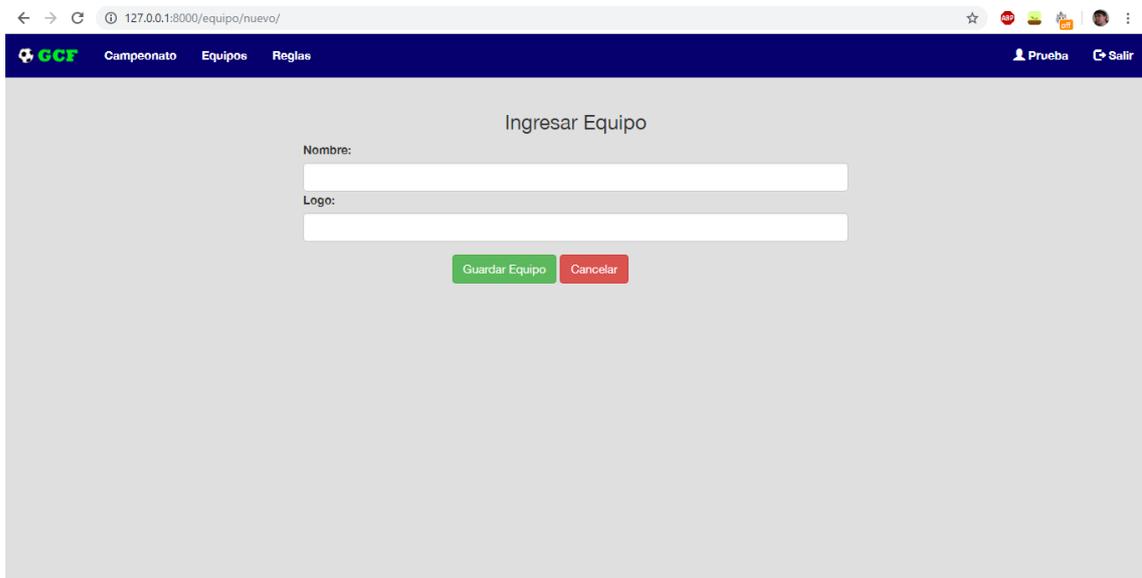


The screenshot shows a web browser window with the URL `127.0.0.1:8000/campeonato/`. The page header is identical to the previous screenshot. The main content area is divided into two sections. On the left, there is a large image of the 'CAMPEONATO NACIONAL' logo, which includes the ANFP logo and the Scotiabank logo. Below the image, the text 'CampeonatoPru' is displayed with a trash icon, and a green 'Comenzar' button is present. A note at the bottom of this section reads: '*Se deben tener mínimo 8 equipos para comenzar'. On the right, there is a table titled 'Equipos Participantes' with a blue header. The table has two columns: '#' and 'Equipos Participantes'. The first row shows '0' in the '#' column and a yellow 'Crear Equipos' button in the 'Equipos Participantes' column.

Fuente: Elaboración propia

Anexo I: Crear equipo

Ilustración 10. Crear equipo

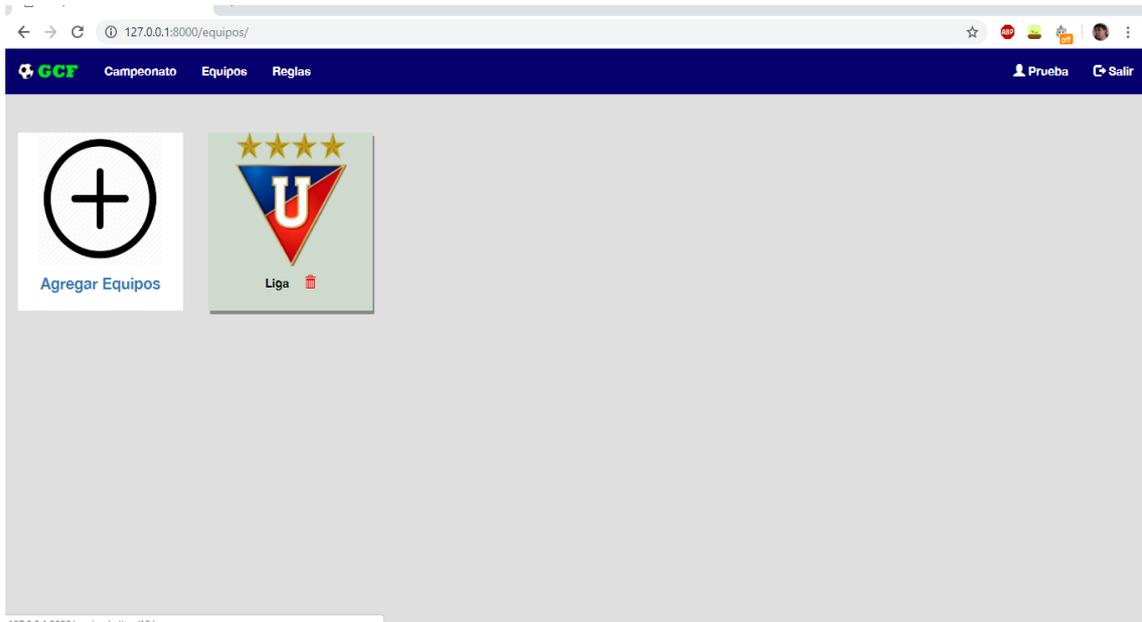


The screenshot shows a web browser window with the URL `127.0.0.1:8000/equipo/nuevo/`. The page has a dark blue header with the GCF logo and navigation links for 'Campeonato', 'Equipos', and 'Reglas'. On the right side of the header are links for 'Prueba' and 'Salir'. The main content area is titled 'Ingresar Equipo' and contains a form with two input fields: 'Nombre:' and 'Logo:'. Below the form are two buttons: a green 'Guardar Equipo' button and a red 'Cancelar' button.

Fuente: Elaboración propia

Anexo J: Editar y eliminar equipo

Ilustración 11. Editar y eliminar equipo



Fuente: Elaboración propia

Anexo K: Listado de partidos

Ilustración 12. Listado de partidos

#	Final	Acciones
	*Se debe jugar el partido del 3er lugar	
#	Partido 3 lugar	Acciones
	*Se deben jugar todos los partidos de 2 fase	
#	Partidos Semifinal	Acciones
	*Se deben jugar todos los partidos de 1 fase	
#	Partidos 1 Fase	Acciones
1	Fuerza Amarilla VS Liga	Jugar
2	Fuerza Amarilla VS Emelec	Jugar
3	Fuerza Amarilla VS Barcelona	Jugar
4	D Cuenca VS El Nacional	Jugar
5	Independiente VS Fuerza Amarilla	Jugar

Fuente: Elaboración propia

Anexo L: Jugar partido

Ilustración 13. Jugar partido

Emelec VS Barcelona

Emelec
Ingrese el número de goles

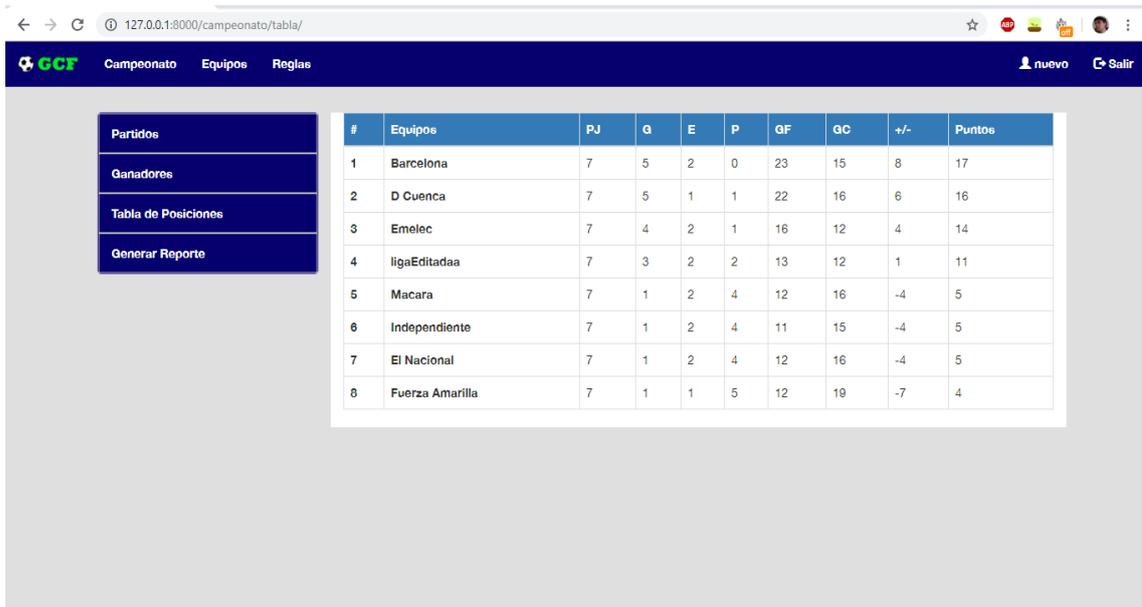
Barcelona
Ingrese el número de goles

Guardar Cancelar

Fuente: Elaboración propia

Anexo M: Tabla de posiciones

Ilustración 14. Tabla de posiciones

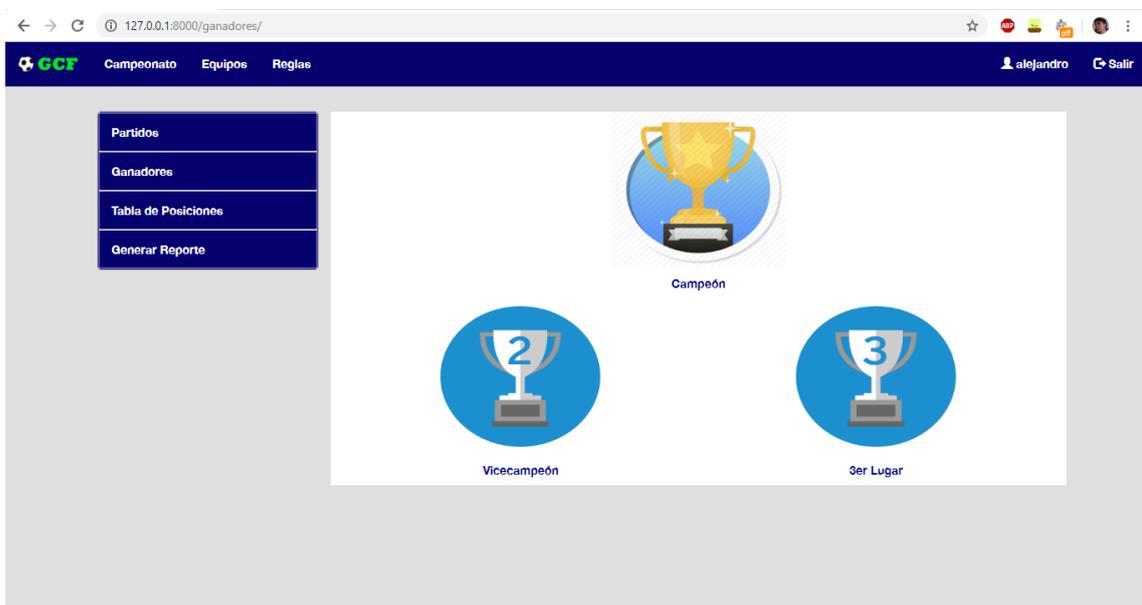


#	Equipos	PJ	G	E	P	GF	GC	+/-	Puntos
1	Barcelona	7	5	2	0	23	15	8	17
2	D Cuenca	7	5	1	1	22	16	6	16
3	Emelec	7	4	2	1	16	12	4	14
4	ligaEditadaa	7	3	2	2	13	12	1	11
5	Macara	7	1	2	4	12	16	-4	5
6	Independiente	7	1	2	4	11	15	-4	5
7	El Nacional	7	1	2	4	12	16	-4	5
8	Fuerza Amarilla	7	1	1	5	12	19	-7	4

Fuente: Elaboración propia

Anexo N: Ganadores no definidos

Ilustración 15. Ganadores no definidos



Fuente: Elaboración propia

Anexo Ñ: Ganadores definidos

Ilustración 16. Ganadores campeonato



Fuente: Elaboración propia