


UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA CENTRO REGIONAL DE ZACAPA FACULTAD DE : INGENIERIA EN SISTEMAS DE INFORMACION ASIGNATURA : ANALISIS DE SISTEMAS II CICLO : 7mo. CATEDRATICO : ESTUARDO VILLATORO DURACION : 2 DIAS		 CALIFICACION ABSOLUTA : 20 RELATIVA : 100 FECHA : 25/09/2021
EXAMEN: <input type="checkbox"/> 1er. PARCIAL <input checked="" type="checkbox"/> 2do. PARCIAL <input type="checkbox"/> FINAL <input type="checkbox"/> RECUPERACION <input type="checkbox"/> EXTRAORDINARIO <input type="checkbox"/> OTRO		
NOMBRE: Jordy Alexander Vega Aldana		CARNE : 1190-18-9769
NOTAS E INSTRUCCIONES		ESCALA DE PUNTUACION Y VALORACION INDICADA

La liga de fútbol profesional guatemalteca ha decidido automatizar sus sistemas creando una base de datos para guardar la información de los partidos que se juegan en la liga.

Se desea guardar en primer lugar los datos de los jugadores. De cada jugador se quiere guardar el nombre, fecha de nacimiento y posición en la que juega (portero, defensa, etc.). Cada jugador tiene un código de jugador que lo identifica de manera única.

De cada uno de los equipos de la liga es necesario registrar el nombre del equipo, nombre del estadio en el que juega, el aforo que tiene, el año de fundación del equipo y la ciudad de la que es el equipo. Cada equipo también tiene un código que lo identifica de manera única. Un jugador solo puede pertenecer a un único equipo.

De cada partido que los equipos de la liga juegan hay que registrar la fecha en la que se juega el partido, los goles que ha metido el equipo de casa y los goles que ha metido el equipo de fuera. Cada partido tendrá un código numérico para identificar el partido.

También se quiere llevar un recuento de los goles que hay en cada partido. Se quiere almacenar el minuto en el que se realizar el gol y la descripción del gol. Un partido tiene varios goles y un jugador puede meter varios goles en un partido.

Por último, se quiere almacenar, en la base de datos, los datos de los presidentes de los equipos de fútbol (DPI, nombre, apellidos, fecha de nacimiento, equipo del que es presidente y año en el que fue elegido presidente). Un equipo de fútbol tan sólo puede tener un presidente, y una persona sólo puede ser presidente de un equipo de la liga.

SE necesita una aplicación multiplataforma para informar a la afición sobre todos estos datos y dar seguimiento a todo lo mediático de la Liga

Para esto debe presentar un análisis de un sistema haciendo énfasis en los temas vistos en clase:

- En que arquitectura lo desarrollara y porque
- Cual es su propuesto de UI y UX
- Que tipos de usabilidad aplicó
- Y como garantizara la Calidad de la Aplicación Desarrollada.
-

Como siempre deberá presentar su propuesta de la forma que usted mejor lo considere para poder defenderla.

Análisis y desarrollo de la Aplicación

Jordy Alexander Vega Aldana 1190-18-9769

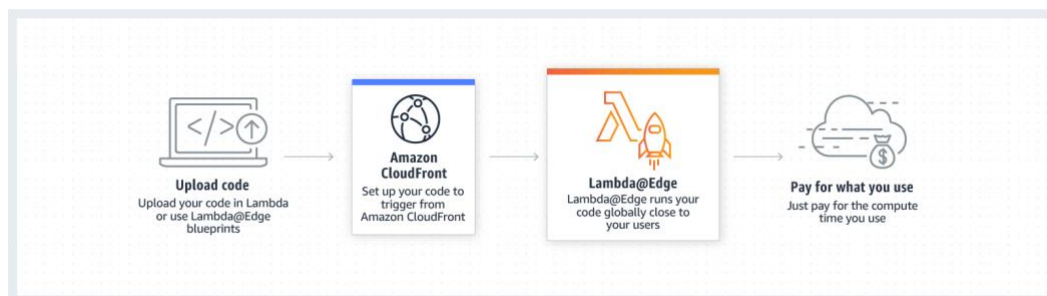
Link de github: <https://github.com/JordyV3/segundo-parcial-analisis>

Para el desarrollo de la aplicación multiplataforma para la liga nacional de futbol se buscará realizar una APP WEB, basada en reactjs pasada en componentes la cual se puede integrar a un React Native para llevarla a dispositivos Android e IOS, todo basado en componentes reutilizables y escalables.

Tecnologías para utilizar para el desarrollo

- Reactjs
- React Native
- Nodejs
- MongoDB
- Express
- Y servicios serverless AWS Lambda

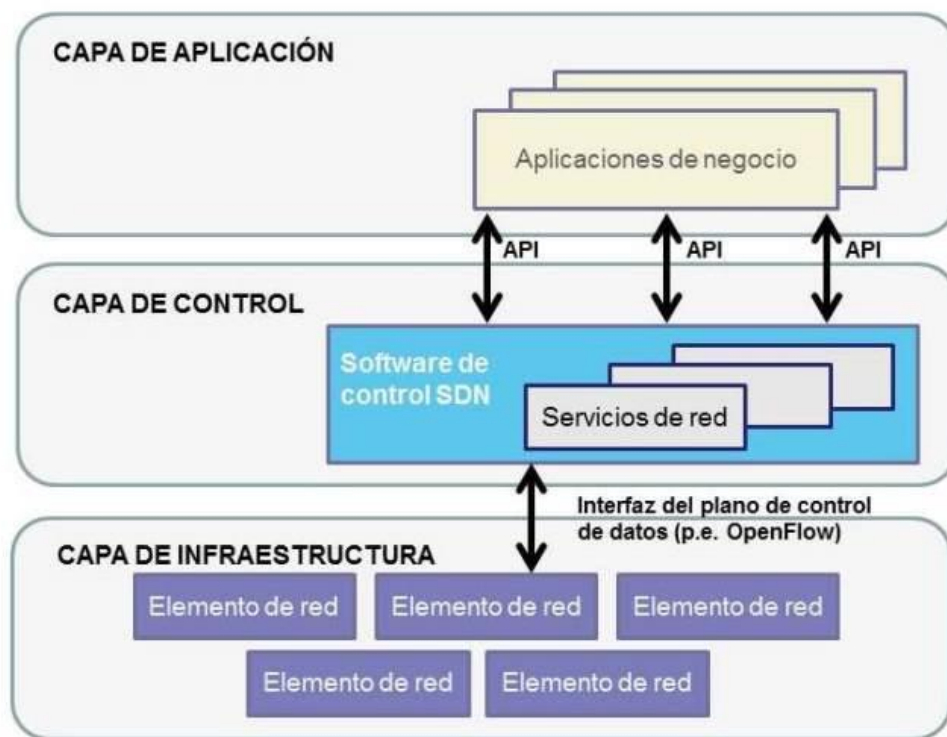
Todas estas tecnologías se estarán utilizando para alcanzar un desarrollo he implementación de la manera mas rápida y eficiente posible, para conseguir un despliegue y usabilidad en el menor tiempo posible por parte de los usuarios.



Arquitectura para desarrollar el sistema de liga

Se utilizará la arquitectura en capas, ya que es el mas común de todos y con un mayor nivel de velocidad para el desarrollo de este. Los patrones de arquitectura en capas son patrones de n niveles donde los componentes están organizados en capas horizontales. Este es el método tradicional para diseñar la mayoría de los programas informáticos y está destinado a ser auto-independiente. Esto significa que todos los componentes están interconectados, pero no dependen unos de otros. Cada capa del patrón de arquitectura en capas tiene un papel y una responsabilidad específicos dentro de la aplicación. Por ejemplo, una capa de presentación se encargaría de manejar toda la interfaz de usuario y la lógica de comunicación del navegador, mientras que una capa empresarial se encargaría de ejecutar las reglas empresariales específicas asociadas a la solicitud.

Una de las características poderosas del patrón de arquitectura en capas es la separación de las preocupaciones entre los componentes. Los componentes dentro de una capa específica se ocupan sólo de la lógica que pertenece a esa capa.



Ventajas del modelo de software basado en capas

- Alta COM probabilidad porque los componentes pertenecen a capas específicas de la arquitectura, otras capas pueden ser burladas o desviadas, haciendo que este patrón sea relativamente fácil de comprobar.
- Alta facilidad de desarrollo porque este patrón es muy conocido y no es excesivamente complejo de implementar, además la mayoría de las empresas desarrollan aplicaciones separando conjuntos de habilidades por capas, este patrón se convierte en una elección natural para la mayoría de los desarrollos de aplicaciones empresariales.
- Mantenible.
- Fácil de asignar «roles» separados.
- Fácil de actualizar y mejorar las capas por separado.

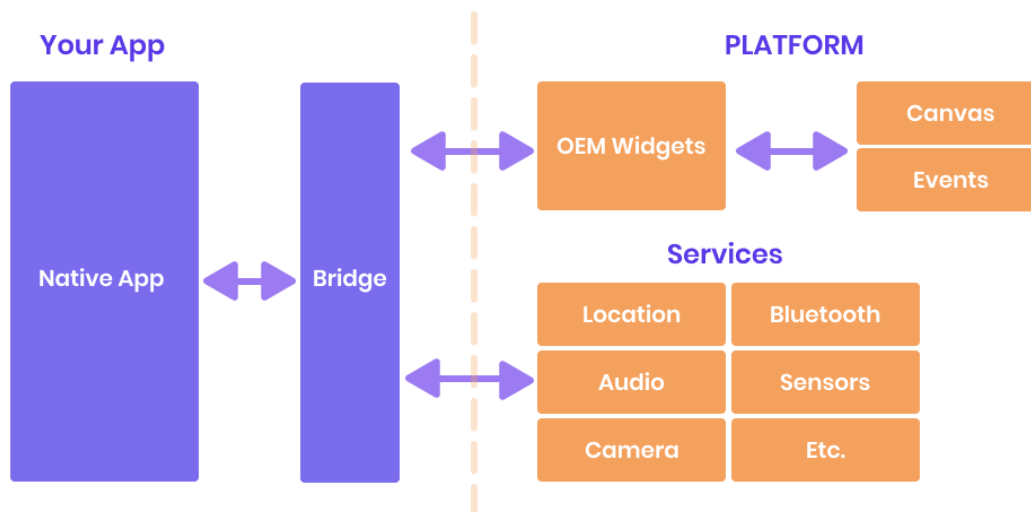
Casos de uso del patrón

- Aplicaciones estándar de línea de negocios que hacen más que sólo operaciones CRUD
- Nuevas aplicaciones que necesitan ser construidas rápidamente
- Equipos de desarrolladores inexpertos que aún no entienden otras arquitecturas
- Aplicaciones que requieren normas estrictas de mantenimiento y COM probabilidad.

Arquitectura del desarrollo de software multiplataforma con reactjs

“REACT JS: la nueva tendencia en Aplicaciones web y multiplataforma”

se busca recopilar información a través de los estudios que se basaron y aplicaron técnicas similares a lo planteado; esos estudios permitieron sintetizar un poco más el concepto que tenía cada autor, así mismo las controversias que se presentaron, referente a React JS ya que muchos lo definen como una biblioteca de JavaScript, que tiene dominio y rendimiento sobre el DOM virtual, que es un framework, entre otros, por otro lado, existen autores que lo definen como todo lo contrario, las ventajas y desventajas teniendo en cuenta las experiencias frente al resultado obtenido, el control que se dio en los datos que recopilaron, entre otros sucesos.



Usabilidad

La necesidad de desarrollar aplicaciones informáticas que faciliten la realización de tareas a los usuarios se ha convertido en un factor determinante para la mayoría de los diseñadores/desarrolladores Web. Muchas organizaciones han incluido en sus proyectos requisitos de usabilidad en sus especificaciones de requisitos de software, pues han identificado la importancia que representa desarrollar productos "usables" que los ayuden a atraer la mayor cantidad de usuarios a sus aplicaciones.

USABILIDAD & EXPERIENCIA DE USUARIO



Usabilidad aplicada

- **Siempre visible para los usuarios:** Tener la plataforma siempre en línea a través de servicios en la nube.
- **Darle libertad al usuario:** El usuario tiene todo el control de lo que va a querer ver en la web y en el sistema de administración.
- **Diseño amigable:** Diseño minimalista y lo más simple posible para que el usuario no se aburra con la interfaz.
- **Libre de errores:** Que toda la plataforma esté libre de errores por consola y que todas las funcionalidades estén en óptimas condiciones.
- **Eficiencia y flexibilidad:** Tener una buena eficiencia de la plataforma web para que los usuarios tengan la información siempre lista y a la hora, para consultar alineaciones resultados de sus equipos favoritos.

Diagrama entidad relación (Base de datos)

Jordy Alexander Vega Aldana 1190-18-9769

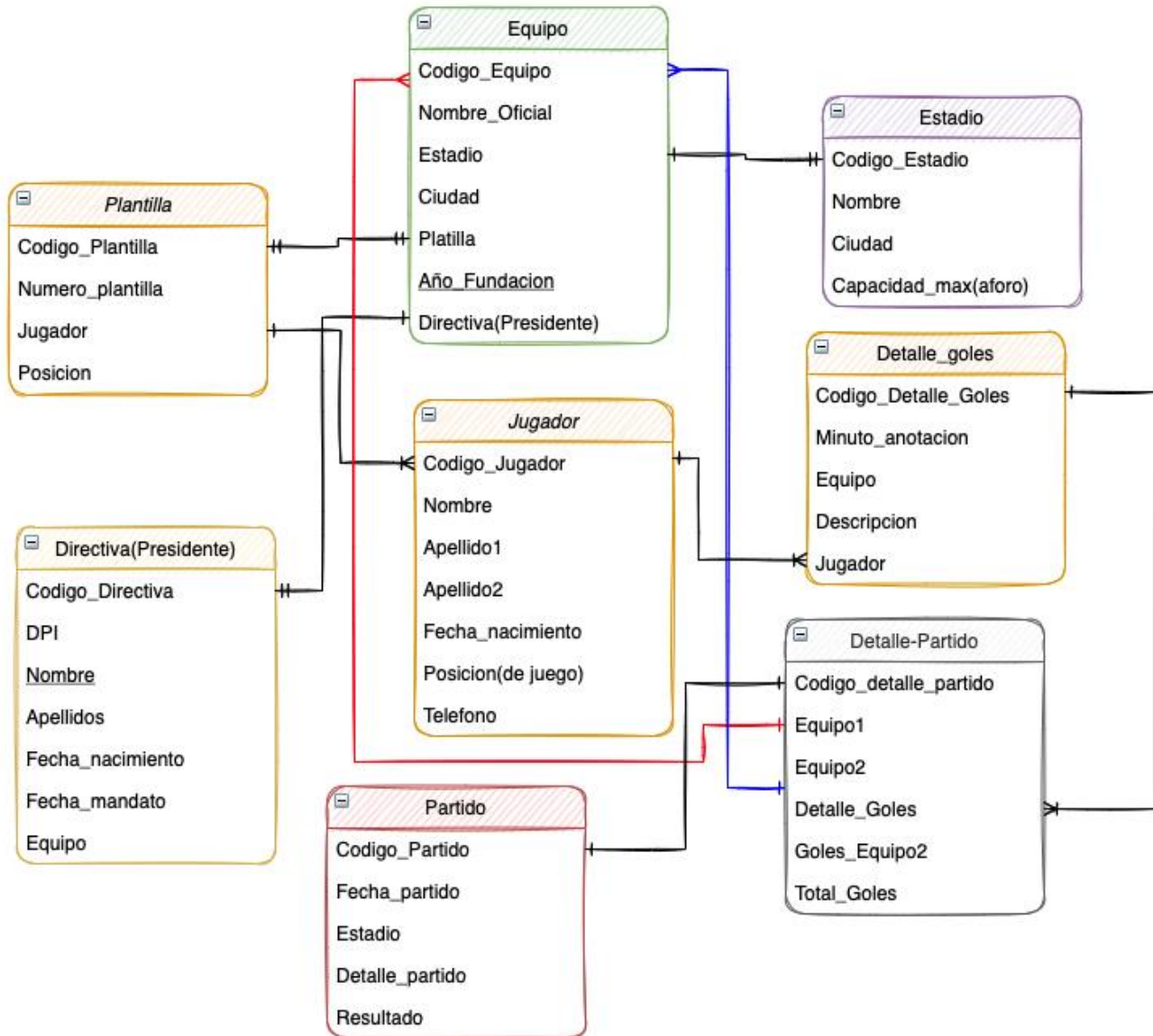
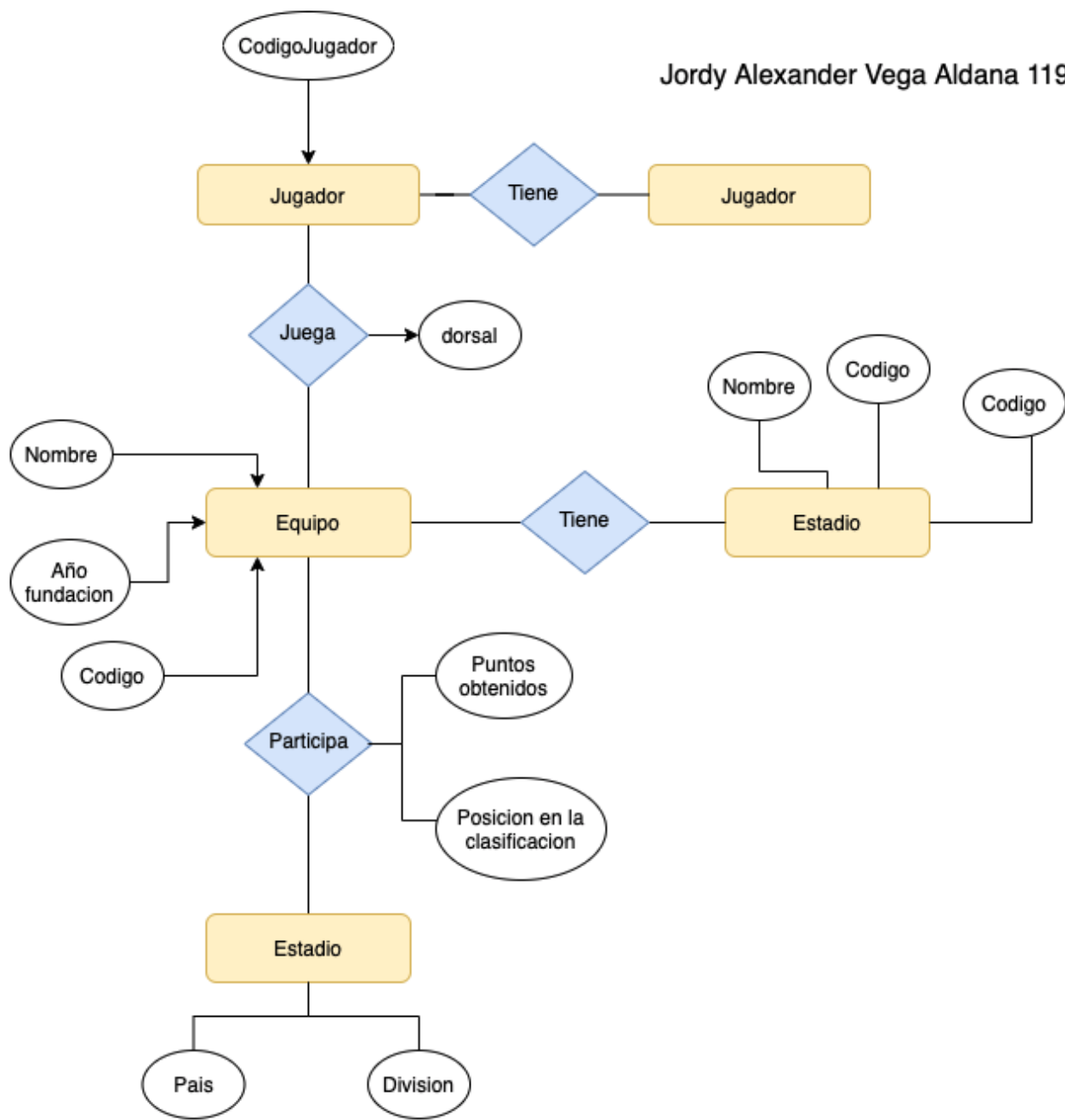


Diagrama de flujo

Jordy Alexander Vega Aldana 1190-18-9769



Interfaz de usuario UI UX

Un diseño centrado en el usuario (DCU) se basa en los fundamentos del diseño UX/UI, creando para y por el usuario, en contraposición a lo que podría ser un diseño centrado en la tecnología o uno centrado en la creatividad u originalidad. Te sugerimos adentrarte en las nuevas tendencias UX/UI y probar los mejores programas de UX para diseñadores web, si quieres iniciar tu camino como diseñador de buenas experiencias.

Demo de la interfaz:

<https://www.figma.com/proto/jjVPcpOTotKggIecwXThIu/Interfaz-usuario-Analisis-Jordy-Vega-1190-18-9769?node-id=1%3A2>

Home Partidos Informacion

 La Liga Guatemala



Futbol Guatemalteco

Vive la pacion del futbol....

Empieza a seguir



Cuando quieras y donde quieras

Revisa lineasiones y resultados en donde gustes con la APP movil.

Los mejores equipos



Principales equipos




Patrocinadores




700k

 Youtube Subscribers

2.4m

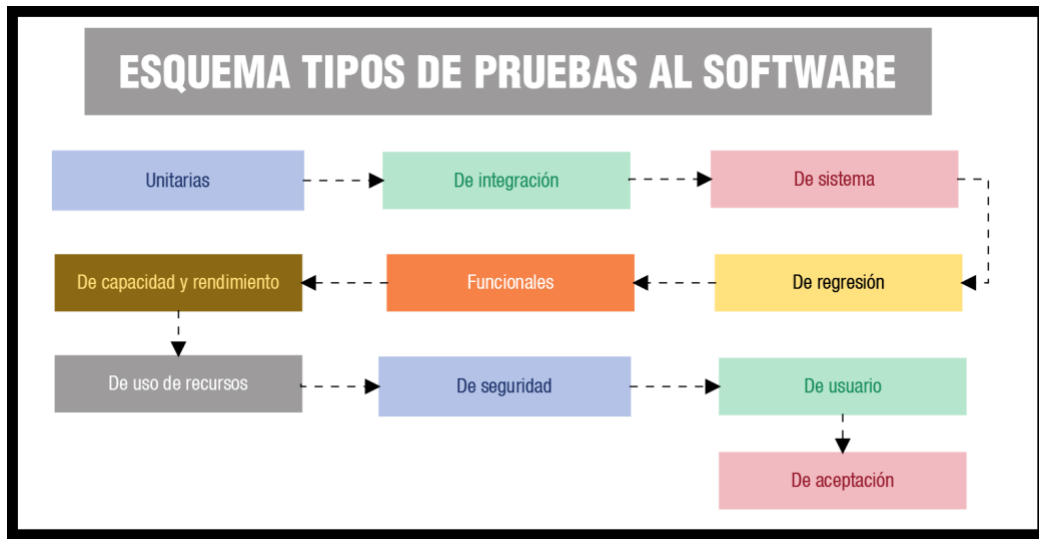
 Instagram Followers

100k

 Dribbble Shot Likes

Calidad de la Aplicación para desarrollar

Se buscará tener la mejor calidad posible en el desarrollo de la aplicación para entregar al cliente un producto 100% funcional y libre de errores para que todos puedan disfrutar del fútbol nacional.



Esto se logrará haciendo las siguientes pruebas a la Aplicación.

- **Pruebas funcionales:** Las pruebas funcionales se llevan a cabo para comprobar las características críticas para el negocio, la funcionalidad y la usabilidad. Las pruebas funcionales garantizan que las características y funcionalidades del software se comportan según lo esperado sin ningún problema.
- **Pruebas unitarias:** Las pruebas de integración implican probar diferentes módulos de una aplicación de software como grupo. Una aplicación de software se compone de diferentes submódulos que trabajan juntos para diferentes funcionalidades
- **Pruebas no funcionales:** Las pruebas no funcionales son como pruebas funcionales; sin embargo, la principal diferencia es que esas funciones se prueban bajo carga para el rendimiento de los observadores, fiabilidad, usabilidad, escalabilidad, etc.
- **Performance Testing:** Las pruebas de rendimiento son un tipo de pruebas no funcionales, realizadas para determinar la velocidad, estabilidad y escalabilidad de una aplicación de software.