

Universidad Mariano Gálvez de Guatemala

Carrera: Ingeniería en Sistemas de Información y ciencias de Computación

Catedrático. Ing. Michael Rodolfo Asturias López

Curso: Aseguramiento de la Calidad del Software

Sección: "B"

Gestión de Quiniela a través de interfaz API Rest

Byron Estuardo Román Chanquin 0900-15-16237

Coralía Elizabeth Cuellar de la Cruz 0900-06-2772

## Índice

1 introducción .....	3
1.1 Proposito.....	4
1.2 Objetivos.....	4
1.3 Alcance.....	4
1.3.1 Pantallas de APIS rest.....	5
1.3.2 Diccionario de datos.....	5
1.3.3 Reglas de Quiniela.....	6
1.4 Tecnología utilizada.....	6
2. Planificación y Costes.....	7
2.1 Planificación.....	7
2.2 Recursos y costos económicos.....	8
3. Descripción general.....	9
3.1 Requerimientos funcionales.....	10
4. Especificación.....	16
5. Creación de base de datos.....	24
6. Código de la aplicación en PHP.....	24

## **1. Introducción**

El Presente proyecto es un software que permita administrar un sistema de apuestas o bien para distracción del usuario donde se pone en práctica todo lo que hemos aprendido en el semestre sobre el Aseguramiento y calidad del software. El proyecto es una forma de definir en detalle los entregables guías de cada área de conocimiento. Es resultado de este proyecto nos lleva a la realización de una aplicación que sea capaz de ofrecer satisfacción a los usuarios tanto para realizar apuestas como para los que solo quieren pasar un rato ameno con lo que más les gusta. Se propone que lo anterior esté soportado en una herramienta web que permita automatizar el almacenamiento de los indicadores, y que esta sea amigable al usuario.

Más del 40% de las personas se considera fan del fútbol, esto hace que sea el deporte más famoso del mundo, por lo tanto, los usuarios pueden aprovechar el uso de la aplicación.

El objetivo de este proyecto es crear una aplicación con api rest para la gestión de una de quinielas de fútbol. Esta aplicación utilizará una base de datos para el almacenamiento de los datos, para resguardar la integridad, la seguridad y evitar la corrupción de estos. Las contraseñas de los usuarios que tendrá la aplicación se guardarán en la base de datos.

Esta aplicación también ayudará a manejar de la mejor manera la gestión de las apuestas en las quinielas mediante las diversas pantallas las cuales guiarán a los usuarios para que puedan visualizar la información que necesiten.

## 1.1 Propósito

Que los participantes se integren de forma que puedan ganar la mayor parte de los premios, según el mayor número de aciertos y puedan pasar un rato agradable con la experiencia de la aplicación.

## 1.2 Objetivos

Es crear la aplicación con interfaz api rest para gestor de quiniela usando metodologías ágiles aprovechando la tecnología Open Source actuales y hacer que las funcionalidades sean amigables para los usuarios.

También poder poner en práctica la utilización de metodologías lo cual nos lleva a crear software de mejor calidad y seguridad y que el cliente se sienta satisfecho y seguro al usar nuestra aplicación.

## 1.3 Alcances

Gracias al alcance del internet el consumo de este juego se ha mantenido en buena medida y este incrementa para la época mundialista realizada cada 4 años. La aplicación de quiniela ofrece un conjunto de información y funcionalidades las cuales se pueden consultar y usar en el momento que lo requiera el usuario.

Primero se crea el diseño de la aplicación en api rest, después se desarrollarán las funcionalidades, se configuran todos los módulos necesarios para el buen funcionamiento de esta. También se diseñan y se crean las tablas de la base de datos para el correcto funcionamiento de la aplicación.

Se crean dos tipos de usuarios: administrador y usuario normal pero que podrán jugar cada usuario tendrá acceso a diferentes pantallas según el nivel de acceso que tengan.

Mediante el sistema de validación se puede participar en los concursos por los métodos de apuesta, los resultados de los partidos y la clasificación de los equipos del mundial de futbol.

### 1.3.1 Pantallas de API REST

Los usuarios de tipo administrador tienen permisos para gestionar la aplicación, y podrá realizar las modificaciones que considere necesarias. Para lo cual crearán las siguientes pantallas para los usuarios:

- Una pantalla de alta de un usuario para poder crear un nuevo usuario en el sistema el cual tendrá el rol de jugador.
- Una pantalla para poder realizar modificaciones a los usuarios.
- Una pantalla que se usará para ver los resultados de los partidos.
- Se creará una pantalla para poder ver qué apuestas que tienen asignado a un jugador para hacer la apuesta.
- La fecha a partir de la cual se crea la quiniela para apostar.
- Una pantalla donde se podrán consultar los aciertos que tengan cada uno de los jugadores.
- Una pantalla cada jugador pueda realizar sus predicciones de los juegos que se realizaran.
- Una pantalla que se utilizará poder realizar la modificación de las apuestas.
- Se creará una pantalla para poder consultar qué partidos se juegan en cada jornada.
- Se creará una página donde el administrador podrá definir reglas para la apuesta generada.
- Se crea una pantalla donde su puede visualizar en que clasificación quedaron los equipos, ejemplo, primera, segunda división.

Para facilitar la gestión de gastos e ingresos se creará diversas pantallas:

- Una pantalla que se utilizará para administrar los pagos de los jugadores por jugar en la quiniela.
- Un usuario que entra a la liga como jugador es un usuario que tiene permisos para apostar, pero no para gestionar la aplicación.
- Se creará las siguientes pantallas para los usuarios de tipo jugador:
  - Se creará una pantalla para que el jugador pueda visualizar sus aciertos, los aciertos de otros jugadores y qué jugador ha hecho más aciertos.
  - Una pantalla de la liga general para que el jugador designado pueda hacer la apuesta que corresponde.
  - Se creará una pantalla de los partidos, donde el jugador podrá ver los partidos que se juegan en una jornada.
  - Una pantalla donde el jugador podrá modificar sus datos personales.
  - Se creará una pantalla de juegos donde muestra los resultados para pueda ver los juegos jugados, partidos ganados cómo han ido evolucionando los aciertos de los jugadores en cada jornada.
  - Una pantalla con la clasificación de los equipos de la primera y segunda división en el mundial de futbol.

### 1.3.2 Diccionario de datos

Nombre

Correo

Quinielas

Partidos

Países

Grupos

Password

Estado

Valor

Configuración

Código

Estructura  
usuario  
Fecha  
Ubicación  
Jugador  
Goles  
Selección  
Grupo  
Ganador  
Juego  
Estatus  
Puesto  
Premio  
Ciudad

#### 1.3.3 Reglas de Quiniela

- Solo usuario registrado podrá ingresar a la aplicación
- La liga solo permitirá 3 primeros lugares
- Al existir un empate en el primer lugar, el premio se dividirá en forma equitativa.
- Al existir empate en el tercer lugar, el premio se dividirá de forma equitativa.
- Al existir empate en el segundo lugar, el premio se dividirá de forma equitativa.

#### 1.4 Tecnología Utilizada

La aplicación es web ya que es esto no requiere que sea instalada en el dispositivo de los usuarios, únicamente es necesario que tengan un navegador web y conexión a internet.

Se ha usado el lenguaje php para crear las Api rest para que las Apis sean dinámicas, se utiliza este lenguaje porque es gratis seguro y sencillo, también para la gestión de base de datos se utiliza MySQL server, porque es de acceso libre y es rápido el procesamiento de la información.

## 2 Planificación y Costes

### 2.1 Planificación

El proyecto se ha dividido en las siguientes fases:

- Estudio Previo: Obtener información y decidir que tecnologías, lenguajes se van a usar para desarrollar las API rest.
- Instalar las aplicaciones necesarias para poder crear la aplicación.
- Definición del objetivo: Definir los objetivos y las funcionalidades de las API rest.
- Especificación de utilidades: Especificación de los scripts para la gestión de la aplicación.
- Especificación de funciones: Especificación detallada de las funcionalidades de las APIS.
- Informe previo: Redactar el informe previo con la descripción de los objetivos del proyecto, explicar cada uno de los objetivos alcanzados y planificación del trabajo a realizar para finalizar el proyecto.
- Instalación y configuración del Servidor: Instalar y configurar el ordenador que se va a usar para instalar la aplicación web.
- implementación: a través de manuales se realizan los pasos para la implementación de la aplicación web.
- Pruebas: estas pruebas de la aplicación sirven para garantizar la estabilidad de funcionamiento y comprobación de compatibilidad de la aplicación con varios navegadores web.
- Documentación: Se crea la guía de configuración del servidor e instalación del sistema.



## 2.2 Recursos y costos económicos

En este apartado se muestra el estudio económico del proyecto. Los costos se dividen en tres partes: los costos de hardware, de software y de recursos humanos.

Hardware: es importante tener acceso a un ordenador con conexión a Internet para usarlo como servidor, por lo cual se utilizará un servidor en la nube, por lo tanto, es necesario un dominio de Internet para que se pueda acceder a la aplicación web desde Internet.

Software: El sistema operativo Windows, todas las aplicaciones que se han usado para desarrollar la aplicación (PHP, XAMP, MySQL, Google, Chrome) y las aplicaciones que aseguran el correcto funcionamiento de la aplicación web (postman, MVC). Como todas las aplicaciones son software libre por lo tanto no tienen ningún costo.

Recursos Humanos: En la creación de este proyecto donde han participado dos personas con diferentes roles, un programador, un diseñador, un administrador de sistemas, un analista y el jefe de proyecto.

Se ha supuesto que se ha trabajado 4 horas diarias. Los costos de recursos humanos se pueden ver en la siguiente tabla:

Tarea	Horas	Perfil	Costo/hora	Total
Estudio previo	12	Jefe de proyecto	150.00	Q.1800.00
Definición de objetivos	6	Analista	130.00	Q. 780.00
Especificación de utilidades	8	Analista	130.00	Q.1040.00
Especificación de funciones	12	Analista	130.00	Q.1560.00
Informe previo	12	Programador	150.00	Q.1920.00
Instalación y configuración del Servidor	12	Administrador de sistemas	150.00	Q.1800.00
implementación	72	programador	120.00	Q.8640.00
Pruebas	36	programador	120.00	Q.4320.00
Documentación	72	programador	120.00	Q.8640.00
Total	242			Q.30,500.00

### 3 Descripción general

En las funcionalidades del gestor de quinielas Hay dos tipos de usuarios que pueden identificarse en la API rest:

Los usuarios administrador y jugador. Un usuario no identificado sólo puede acceder a la página principal (la página de identificación al sistema) en la cual se tendrá que registrar para tener acceso a las demás pantallas.

El usuario administrador es el único que puede gestionar los datos de la aplicación. El tipo de usuario con nivel de administrador tiene permisos para controlar las jornadas que se van a jugar, apuestas, realizar altas y bajas de usuarios etc. Para que la integridad de la aplicación prevalezca el administrador no puede apostar, sólo puede hacer la liga general en el caso de que el jugador no la ha ya hecho. El administrador y jugador son perfiles totalmente distintos.

El tipo de usuario normal que no es el administrador se ha creado porque se necesitaba un perfil de usuario que pueda apostar o bien solo participar en un juego solo por diversión. Un jugador puede consultar los partidos de la liga creada, las apuestas de otros jugadores, apostar etc.

Para seguridad del usuario administrador debe loguearse en la aplicación. El Administrador puede crear otros administradores y jugadores. También existe otro tipo de usuario y es el que participa en las apuestas de las ligas, el usuario administrador tiene otros accesos. Este perfil fue necesario definirlo porque se requería de un usuario para gestionar varios procesos de la aplicación, tales como la programación de las ligas, realizar las apuestas en la quiniela, por lo tanto el usuario administrador es quien tiene control total sobre la aplicación.


### 3.1 Requerimientos del usuario


Especificación de Requerimientos Funcionales						
<b>Id del Requisito</b>	RF01					
<b>Nombre del Requisito</b>	Gestion de modulo registro de usuarios a la pagina					
<b>Componente</b>	API rest, visual studio, MySQL server					
<b>Descripcio del Requisito</b>	El usuario podra registrarse enviando sus datos (como nombre, apellido y direccion de correo)					
<b>Caracteristica</b>	El sistema permitira que los usuarios se puedan registrar					
<b>Prioridad</b>	X	Alta/Esencial		Media/Deseado		Baja/Opcional
<b>Restricciones</b>	si no esta registrado no podra ingresar					


Especificación de Requerimientos Funcionales						
<b>Id del Requisito</b>	RF02					
<b>Nombre del Requisito</b>	Gestion Modulo creacion de ligas					
<b>Componente</b>	API rest, visual studio, MySQL server					
<b>Descripcio del Requisito</b>	El usuario podra crear ligas que necesite debe indicar si es de apuesta o de diversion					
<b>Caracteristica</b>	El sistema permitira al usuario administrar crear ligas					
<b>Prioridad</b>	X	Alta/Esencial		Media/Deseado		Baja/Opcional
<b>Restricciones</b>	si no es administrador no puede crear ligas					


Especificación de Requerimientos Funcionales						
<b>Id del Requisito</b>	RF03					
<b>Nombre del Requisito</b>	Invitacion a quiniela					
<b>Componente</b>	API rest, visual studio, MySQL server					
<b>Descripcio del Requisito</b>	El usuario podra enviar invitacion a todos los usuario					
<b>Caracteristica</b>	La invitación se le puede enviar a cualquier usuario por medio de correo tanto registrado como no registrado					
<b>Prioridad</b>	X	Alta/Esencial		Media/Deseado		Baja/Opcional
<b>Restricciones</b>						


Especificación de Requerimientos Funcionales						
<b>Id del Requisito</b>	RF04					
<b>Nombre del Requisito</b>	Gestion de modulo de login					
<b>Componente</b>	API rest, visual studio, MySQL server					
<b>Descripcio del Requisito</b>	El sistema le permita ingresar con usuario y contraseña					
<b>Caracteristica</b>	El sistema permitira el ingreso con credenciales correctas					
<b>Prioridad</b>	X	Alta/Esencial		Media/Deseado		Baja/Opcional
<b>Restricciones</b>						


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF05			
<b>Nombre del Requisito</b>	Gestion de modulo Administrador			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sistema le permita ingresar con usuario y contraseña			
<b>Caracteristica</b>	El sistema debe permitir el ingreso, modificación, lectura y eliminación de de las n pantallas			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF06			
<b>Nombre del Requisito</b>	Gestion de modulo grupos			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sistema debe permitir el ingreso, modificación, lectura y eliminación de registros			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF07			
<b>Nombre del Requisito</b>	Gestion de modulo paises			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sistema debe permitir el ingreso, modificación, lectura y eliminación de estadios			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF08			
<b>Nombre del Requisito</b>	Gestion de modulo para estadios			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sistema debe permitir el ingreso, modificación, lectura y eliminación de estadios			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF09			
<b>Nombre del Requisito</b>	Gestion de modulo para juegos			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sitema debe permitir el ingreso, modificacion, lectura y eliminacion de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF10			
<b>Nombre del Requisito</b>	Gestion de modulo de equipos			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sistema debe permitir el ingreso, modificación, lectura y eliminación de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF11			
<b>Nombre del Requisito</b>	Gestion de modulo estrutura de equipos			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sistema debe permitir el ingreso, modificación, lectura y eliminación de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF12			
<b>Nombre del Requisito</b>	Gestion de modulo prediccion de quinielas			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sitema debe permitir el ingreso, modificacion, lectura y eliminacion de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF13			
<b>Nombre del Requisito</b>	Gestion de modulo de Grupos_paises			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sitema debe permitir el ingreso, modificacion, lectura y eliminacion de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF14			
<b>Nombre del Requisito</b>	Gestion de modulo quinielas			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sitema debe permitir el ingreso, modificacion, lectura y eliminacion de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF15			
<b>Nombre del Requisito</b>	Gestion de modulo premios quinielas			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sitema debe permitir el ingreso, modificacion, lectura y eliminacion de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF16			
<b>Nombre del Requisito</b>	Gestion de modulo tipos quinielas			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sitema debe permitir el ingreso, modificacion, lectura y eliminacion de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial	Media/Deseado	Baja/Opcional
<b>Restricciones</b>				


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF17			
<b>Nombre del Requisito</b>	Gestion de modulo usuarios quiniela			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sitema debe permitir el ingreso, modificacion, lectura y eliminacion de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial		Media/Deseado
<b>Restricciones</b>	Baja/Opcional			


 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF18			
<b>Nombre del Requisito</b>	Gestion de modulo marcado quiniela			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sistema debe mostrar el resultado de los puntos acumulados en la quiniela			
<b>Caracteristica</b>	El sistema tiene que mostrar los puntos por usuario			
<b>Prioridad</b>	X	Alta/Esencial		Media/Deseado
<b>Restricciones</b>	Baja/Opcional			

 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF19			
<b>Nombre del Requisito</b>	Gestion de modulo de ubicación			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sitema debe permitir el ingreso, modificacion, lectura y eliminacion de partidos			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial		Media/Deseado
<b>Restricciones</b>	Baja/Opcional			

 <b>Especificación de Requerimientos Funcionales</b>				
<b>Id del Requisito</b>	RF20			
<b>Nombre del Requisito</b>	Gestion de modulo valor de la liga			
<b>Componente</b>	API rest, visual studio, MySQL server			
<b>Descripcio del Requisito</b>	El sistema debe mostrar el valor por el que se puede crear la liga			
<b>Caracteristica</b>	El administrador puede actualizar, modificar y eliminar los registros			
<b>Prioridad</b>	X	Alta/Esencial		Media/Deseado
<b>Restricciones</b>	Baja/Opcional			

 <b>Especificación de Requerimientos Funcionales</b>	
<b>Id del Requisito</b>	RF21
<b>Nombre del Requisito</b>	Gestion de modulo configuración
<b>Componente</b>	API rest, visual studio, MySQL server
<b>Descripcio del Requisito</b>	El sistema debe permitir la configuración de las tablas maestras
<b>Caracteristica</b>	El sistema tiene que mostrar permitir la configuracion de todos los modulos
<b>Prioridad</b>	X Alta/Esencial      Media/Deseado      Baja/Opcional
<b>Restricciones</b>	

 <b>Especificación de Requerimientos Funcionales</b>	
<b>Id del Requisito</b>	RF22
<b>Nombre del Requisito</b>	Gestion de modulo de reglas de quiniela
<b>Componente</b>	API rest, visual studio, MySQL server
<b>Descripcio del Requisito</b>	El sistema debe mostrar las reglas de la quiniela a los usuarios
<b>Caracteristica</b>	El sistema debe permitir actualizar, modificar y eliminar dichas reglas
<b>Prioridad</b>	X Alta/Esencial      Media/Deseado      Baja/Opcional
<b>Restricciones</b>	

 <b>Especificación de Requerimientos Funcionales</b>	
<b>Id del Requisito</b>	RF23
<b>Nombre del Requisito</b>	Gestion de modulo de calculos de puntos
<b>Componente</b>	API rest, visual studio, MySQL server
<b>Descripcio del Requisito</b>	El sistema debe calcular los premios por los puntos acumulados
<b>Caracteristica</b>	El sistema debe mostrar el calculo de los puntos
<b>Prioridad</b>	X Alta/Esencial      Media/Deseado      Baja/Opcional
<b>Restricciones</b>	



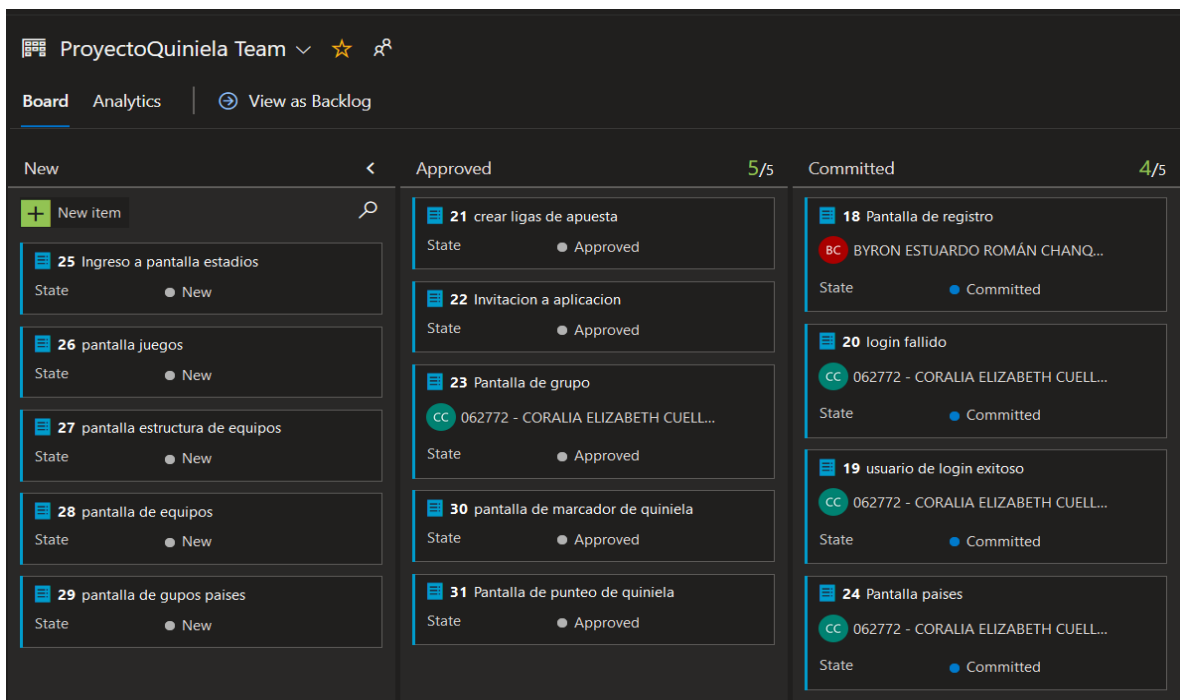
#### 4. Especificación

La especificación nos permite formalizar los requerimientos funcionales definidos anteriormente.

#### Sprint

En este apartado se definen diferentes sprint definidos en azure devops, lo cual nos ayuda a trabajar de manera ágil cada proceso.

En la siguiente imagen se pueden ver las diferentes tareas asignadas en el sistema.



#### Especificaciones de las diferentes asignaciones en azure devops

Lo cual ayudara al desarrollador a realizar el debido proceso para el desarrollo de la aplicación.

ProyectoQuiniela / Boards / Work items

Search

Work items

Recently created

+ New Work Item

↔ Open in Queries

Column Options

↑ Import Work Items

Recycle Bin

Filter by keyword

TypesAssigned toStatesArea

ID	Title	Assigned To	State	Area Path
14	Api rest quiniela	Unassigned	New	ProyectoQuiniela
15	Desarrollo de API	Unassigned	New	ProyectoQuiniela
16	Desarrollo API Web	Unassigned	New	ProyectoQuiniela
17	Desarrollo de base de datos	Unassigned	New	ProyectoQuiniela
18	Pantalla de registro	BYRON ESTUARDO ROMÁ...	Committed	ProyectoQuiniela
19	usuario de login exitoso	062772 - CORALIA ELIZAB...	Committed	ProyectoQuiniela
20	login fallido	062772 - CORALIA ELIZAB...	Committed	ProyectoQuiniela
21	crear ligas de apuesta	Unassigned	Approved	ProyectoQuiniela
22	Invitacion a aplicacion	Unassigned	Approved	ProyectoQuiniela
23	Pantalla de grupo	062772 - CORALIA ELIZAB...	Approved	ProyectoQuiniela
24	Pantalla paises	062772 - CORALIA ELIZAB...	Committed	ProyectoQuiniela

Recently created

↶ Back to Work Items

PRODUCT BACKLOG ITEM 18

18

Pantalla de registro

BC

BYRON ESTUARDO ROMÁN CHANQUIN

0 comments

Add tag

State

Committed

Area

ProyectoQuiniela

Reason

Commitment m...

Iteration

ProyectoQuiniela\Sprint 6\Pantalla grupos

Description

Como: usuario  
Quiero: registrarme en la pagina con mi nombre apellido y correo  
Para: ingresar a la pagina principal.

Acceptance Criteria

Dado: un usuario quiera registrarse  
Cuando: ingrese su nombre, apellido y correo.  
Entonces: le llegara el mensaje de registrado correctamente.

Details

Priority

1

Effort

8

Business Value

Value area

Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 19

19 usuario de login exitoso

062772 - CORALIA ELIZABETH CUELLAR...

0 comments

Add tag

State

Committed

Area

ProyectoQuiniela

Reason

Commitment m...

Iteration

ProyectoQuiniela\Sprint 1

Description

Como Usuario de la aplicacion  
Quiero: que al ingresar mi usuario y contraseña correctos pueda ingresar  
Para: poder ver las diferentes pantallas de la aplicacion.

Acceptance Criteria

Dado: que un usuario quiere loguearse en nuestra aplicacion  
Cuando: ingresa con usuario y contrasella correcta  
Entonces: el usuario podra ingresar exitosamente a la aplicacion

Details

Priority

1

Effort

10

Business Value

Value area

Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 20

20 login fallido

062772 - CORALIA ELIZABETH CUELLAR...

0 comments

Add tag

State

Committed

Area

ProyectoQuiniela

Reason

Commitment m...

Iteration

ProyectoQuiniela\Sprint 3\Pantalla de login

Description

Como: Usuario de la aplicacion  
Quiero: ingrsar el usuario y contraseña incorrecta  
Para: probar que no deje ingresar a la aplicacion con datos falsos

Acceptance Criteria

Dado: que un usuario quiera ingresar a la aplciacion.  
Cuando: con ingrese usuario y contraseña incorrrectos  
Entonces: el sistema le enviara respuesta de usuario o contraseña incorrectas

Details

Priority

2

Effort

7

Business Value

Value area

Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 21

21 crear ligas de apuesta

Unassigned

0 comments

Add tag

State

Approved

Area

ProyectoQuiniela

Reason

Approved by th...

Iteration

ProyectoQuiniela\Sprint 4\Pantalla de creacion de ligas

Description

Como: Usuario Administrador  
Quiero: que el sistema deje crear ligas  
Para: hacer hacer apuestas

Acceptance Criteria

Dado: un usuario admistrador  
Cuando: ingrese nombre, monto, grupo, juegos y estadio  
Entonces: el sistema dara el mensaje de liga creada exitosamente

Details

Priority

1

Effort

6

Business Value

Value area

Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 22\*

22 Invitacion a aplicacion

Unassigned

1 comment

Add tag

State

Approved

Area

ProyectoQuiniela

Reason

Approved by the...

Iteration

ProyectoQuiniela\Sprint 5\Envio correo quiniela

Description

Como: usuario adminitrador  
Quiero: enviar correo a otros usuarios.  
Para: para tener acceso a la aplicacion de la liga

Acceptance Criteria

Dado: que un usuario administrador de la aplicacion  
Cuando: ingrese a la pantalla de invitacion a la liga  
Entonces: el sistema le permitira generar el correo para enviarlo.

Details

Priority

2

Effort

10

Business Value

Value area

Business

Recently created

Back to Work Items

PRODUCT BACKLOG ITEM 23

23 Pantalla de grupo

062772 - CORALIA ELIZABETH CUELLAR...

1 comment

Add tag

State

● Approved

Area

ProyectoQuiniela

Reason

Approved by th...

Iteration

ProyectoQuiniela\Sprint 1

Description

Como : usuarios  
Cuando:quiero consultar los grupos  
Para: hacer pronosticos

Acceptance Criteria

Dado: que un usuario de la aplicacion  
Cuando: ingrese a la pantalla de grupo  
Entonces: el sistema le mostrar los grupos que conforman los equipos.

Details

Priority

2

Effort

9

Business Value

Value area

Business

Recently created

Back to Work Items

PRODUCT BACKLOG ITEM 24

24 Pantalla paises

062772 - CORALIA ELIZABETH CUELLAR...

1 comment

Add tag

State

● Committed

Area

ProyectoQuiniela

Reason

Commitment m...

Iteration

ProyectoQuiniela\Sprint 3

Description

Como: usuario  
Quiero: ingresar a la pantalla de paises  
Para: poder realizar las proyecciones

Acceptance Criteria

Dado: que un usuario de la aplicacion  
Cuando: ingrese a la pantalla de paises  
Entonces: el sistema mostrara los paises participantes.

Details

Priority

2

Effort

9

Business Value

Value area

Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 25

25 Ingreso a pantalla estadios

Unassigned

1 comment

Add tag

State

New

Area

ProyectoQuiniela

Reason

New backlog it ...

Iteration

ProyectoQuiniela\Sprint 3

Description

Como: usuario  
Quiero: ingresar a la pantalla de los estadios  
Para: me muestres los estadios utilizados

Acceptance Criteria

Dado: que un usuario de la aplicacion  
Cuando: ingrese a la pantalla de estadios  
Entonces: el sistema le mostrara el portafolio de estadios

Details

Priority

2

Effort

9

Business Value

Value area

Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 26

26 pantalla juegos

Unassigned

0 comments

Add tag

State

New

Area

ProyectoQuiniela

Reason

New backlog it ...

Iteration

ProyectoQuiniela\Sprint 3

Description

Como: usuario  
Quiero: ingresar a la pagina de juegos  
Para: ver los juegos

Acceptance Criteria

Dado: que un usuario quiere ingresar a la aplicacion  
Cuando: le de click a la pantalla  
Entonce el sistema mostrara todos juegos actualizados

Details

Priority

2

Effort

Business Value

Value area

Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 27

## 27 pantalla estructura de equipos

Unassigned 0 comments [Add tag](#)

State ☒ New Area ProyectoQuiniela  
Reason New backlog it ... Iteration ProyectoQuiniela\Sprint 1\Estrutua de equipos

### Description

Como: usuario  
Quiero: seleccionar ingresar a la pantalla de estructura de equipos  
Para: visualizar si el equipo seleccionado se encuentra en cuartos de final etc

### Details

Priority  
2  
Effort

### Acceptance Criteria

Dado: que un usuario  
Cuando: ingrese a la pantalla de estrucutura de equipos  
Entonces: el sistema muestre los diferentes lugares de los equipos.

Business Value  
  
Value area  
Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 28

## 28 pantalla de equipos

Unassigned 0 comments [Add tag](#)

State ☒ New Area ProyectoQuiniela  
Reason New backlog it ... Iteration ProyectoQuiniela\Sprint 1\Modulo de equipos

### Description

Como: usuario de la aplicacion  
Quiero: poder seleccionar un equipo seleccionando el numero de equipo  
Para: saber que paises estan dentro de ese numero de equipo

### Details

Priority  
2  
Effort

### Acceptance Criteria

Dado: que un usuario ingrese a la pantalla de equipos  
Cuando: seleccione el numero de equipo  
Entonces: el sistema mostrara todos los paises que estan dentro numero de equipo

Business Value  
  
Value area  
Business

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 29

29 pantalla de gupos paises

Unassigned 0 comments [Add tag](#)

State ☒ New

Area ProyectoQuiniela

Reason New backlog it ... Iteration ProyectoQuiniela\Sprint 1\Modulo de equipos

**Description**  
  
Como: usuario registrado  
Quiero: ingresar a la pantalla de grupos paise  
Para: selecciona los grupos con los que quiero participar en el juego

**Details**  
  
Priority 2  
Effort  
  
Business Value  
  
Value area Business

**Acceptance Criteria**  
  
Dado: que el usuario ingresa a la pantalla  
Cuando: ingresas sus datos correctos de usuario y contraseña  
Entonces el sistema mostrara la informacion de los diferentes grupos

Recently created | [Back to Work Items](#)

PRODUCT BACKLOG ITEM 30

30 pantalla de marcador de quiniela

Unassigned 0 comments [Add tag](#)

State ☒ Approved

Area ProyectoQuiniela

Reason Approved by th ... Iteration ProyectoQuiniela\Sprint 6\Pantalla grupos

**Description**  
  
Como: usuario de la pagina  
Quiero: ingresar a la opcion marcador de quiniela  
Para: saber los aciertos obtenidos

**Details**  
  
Priority 2  
Effort  
  
Business Value  
  
Value area Business

**Acceptance Criteria**  
  
Dado: que un usuario del la pagina  
Cuando: ingrese a la pantalla de marcador de quiniela  
Entonces: el sistema le muestre el marcador



Recently created
Back to Work Items

PRODUCT BACKLOG ITEM 31

### 31 Pantalla de punteo de quiniela

Unassigned
0 comments
Add tag

State
Approved
Area
ProyectoQuiniela

Reason
Approved by th ...
Iteration
ProyectoQuiniela\Sprint 6\Pantalla grupos

Description	Details
Como: usuario de la aplicacion	Priority
Quiero: ingresar a la pantalla de punteo de quiniela	2
Para: visuarlizar el punteo de cada participante.	Effort
<b>Acceptance Criteria</b>	Business Value
Dado: que un usuario de la aplicacion	Value area
Cuando: ingresa a la pantalla de punteo	Business
Entonces: el sistema le muestra el punteo de cada participante.	

## 5. Creación de base de datos

Para la crear la interfaz con las Apis se crean las diferentes tablas en la Base de datos quiniela\_mundial, esto con el objeto de almacenar de registra toda la información que se necesita para la creación de las apuestas y los jugadores puedan obtener toda la información posible.

```
drop table if exists ADMINITRADOR;
```

```
CREATE TABLE ADMINISTRADO(
```

```
ID INT(11) not null primary key,
```

```
NOMBRE varchar(80) default null,
```

```
APELLIDO varchar(60) default null,
```

```
CORREO varchar(100) default null,
```

```
PASSWORD varchar(30) default null)
```

```
-- Indices de la tabla `ADMINISTRADORES`
```

```
ALTER TABLE `ADMINISTRADORES`
```

```
ADD PRIMARY KEY (`ID`),  
ADD UNIQUE KEY `ADMINISTRADOR_ID_uindex` (`ID`);
```

-----

```
`CONFIGURACION` (  
  `ID` int(11) NOT NULL,  
  `NOMBRE` varchar(180) DEFAULT NULL,  
  `VALOR` text  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Indices de la tabla `CONFIGURACION`

```
ALTER TABLE `CONFIGURACION`  
  ADD PRIMARY KEY (`ID`),  
  ADD UNIQUE KEY `CONFIGURACION_ID_uindex` (`ID`);
```

-----

-- Estructura de tabla para la tabla `ESTRUCTURAS`

```
DROP TABLE IF EXISTS `ESTRUCTURAS`;  
CREATE TABLE `ESTRUCTURAS` (  
  `ID` int(11) NOT NULL,  
  `NOMBRE` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Indices de la tabla `ESTRUCTURAS`

```
ALTER TABLE `ESTRUCTURAS`  
  ADD PRIMARY KEY (`ID`),  
  ADD UNIQUE KEY `structure_id_uindex` (`ID`);
```

-----

```
-- Estructura de tabla para la tabla `GRUPOS`  
DROP TABLE IF EXISTS `GRUPOS`;  
CREATE TABLE `GRUPOS` (  
  `ID` int(11) NOT NULL,  
  `CODIGO` varchar(5) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-- Indices de la tabla `GRUPOS`  
ALTER TABLE `GRUPOS`  
  ADD PRIMARY KEY (`ID`),  
  ADD UNIQUE KEY `groups_id_uindex` (`ID`);
```

```
-----  
-- Estructura de tabla para la tabla `JUEGOS`  
DROP TABLE IF EXISTS `juegos`;  
CREATE TABLE `juegos` (  
  `id` int(11) NOT NULL primary key,  
  `ESTRUCTURA` int(11) NOT NULL,  
  `FECHA` datetime NOT NULL,  
  `UBICACION` int(11) NOT NULL,  
  `JUGADOR_1` int(11) DEFAULT '99',  
  `JUGADOR_2` int(11) DEFAULT '99',  
  `GOLES_1` int(11) DEFAULT NULL,  
  `GOLES_2` int(11) DEFAULT NULL,  
  `OPCIONES_DE_SELECCION` varchar(30) DEFAULT NULL,  
  `SELECCION_1` varchar(50) DEFAULT NULL,  
  `SELECCION_2` varchar(50) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Indices de la tabla `JUEGOS`

ALTER TABLE `JUEGOS`

ADD PRIMARY KEY (`ID`),

ADD UNIQUE KEY `game\_id\_uindex` (`ID`),

ADD KEY `game\_structure\_id\_fk` (`ESTRUCTURA`),

ADD KEY `game\_locations\_id\_fk` (`UBICACION`),

ADD KEY `game\_countries\_id\_fk` (`JUGADOR\_1`),

ADD KEY `game\_countries2\_id\_fk` (`JUGADOR\_2`);

---

--Estructura de tabla para la tabla `PAISES`

DROP TABLE IF EXISTS `PAISES`;

CREATE TABLE `PAISES` (

  `ID` int(11) NOT NULL,

  `NOMBRE` varchar(40) DEFAULT NULL,

  `ISO` varchar(10) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Indices de la tabla `PAISES`

ALTER TABLE `PAISES`

ADD PRIMARY KEY (`ID`),

ADD UNIQUE KEY `countries\_id\_uindex` (`ID`);

---

-- Estructura de tabla para la tabla `PAISES\_GRUPOS`

DROP TABLE IF EXISTS `PAISES\_GRUPOS`;

CREATE TABLE `PAISES\_GRUPOS` (

  `ID` int(11) NOT NULL,

  `PAIS` int(11) NOT NULL,

  `GRUPO` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Indices de la tabla `PAISES\_GRUPOS`

ALTER TABLE `PAISES\_GRUPOS`

  ADD PRIMARY KEY (`ID`),

  ADD UNIQUE KEY `countries\_groups\_id\_uindex` (`ID`),

  ADD KEY `countries\_groups\_countries\_id\_fk` (`PAIS`),

  ADD KEY `countries\_groups\_groups\_id\_fk` (`GRUPO`);

-----  
DROP TABLE IF EXISTS `QUINIELAS`;

CREATE TABLE `QUINIELAS` (

  `ID` int(11) NOT NULL,

  `NOMBRE` varchar(60) DEFAULT NULL,

  `TIPO\_DE\_QUINIELA` int(11) DEFAULT NULL,

  `DESCRIPCION` varchar(180) DEFAULT NULL,

  `CREADO\_POR` int(11) NOT NULL,

  `FECHA\_DE\_CREACION` date DEFAULT NULL,

  `CODIGO\_COMPARTIR` varchar(12) NOT NULL,

  `GANADOR` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Indices de la tabla `QUINIELAS`

ALTER TABLE `QUINIELAS`

ADD PRIMARY KEY (`ID`),

ADD UNIQUE KEY `quina\_id\_uindex` (`ID`),

ADD UNIQUE KEY `quina\_share\_code\_uindex` (`CODIGO\_COMPARTIR`),

ADD KEY `quina\_quinela\_type\_id\_fk` (`TIPO\_DE\_QUINIELA`),

ADD KEY `quina\_users\_id\_fk` (`CREADO\_POR`),

ADD KEY `quina\_users2\_\_fk` (`GANADOR`);

-----

-- Estructura de tabla para la tabla `QUINIELA\_INVITACIONES`

DROP TABLE IF EXISTS `QUINIELA\_INVITACIONES`;

CREATE TABLE `QUINIELA\_INVITACIONES` (

`ID` int(11) NOT NULL,

`USUARIO` int(11) NOT NULL,

`QUINIELA` int(11) NOT NULL,

`FECHA\_DE\_CREACION` date DEFAULT NULL,

`ESTATUS` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Indices de la tabla `QUINIELA\_INVITACIONES`

ALTER TABLE `QUINIELA\_INVITACIONES`

ADD PRIMARY KEY (`ID`),

ADD UNIQUE KEY `quina\_invitations\_id\_uindex` (`ID`),

ADD KEY `quina\_invitations\_quinela\_id\_fk` (`QUINIELA`),

ADD KEY `quina\_invitations\_users\_id\_fk` (`USUARIO`);

-----

```
-- Estructura de tabla para la tabla `QUINIELA_PREDICCIONES`
DROP TABLE IF EXISTS `QUINIELA_PREDICCIONES`;
CREATE TABLE `QUINIELA_PREDICCIONES` (
  `ID` int(11) NOT NULL,
  `JUEGO` int(11) NOT NULL,
  `QUINIELA` int(11) NOT NULL,
  `USUARIO` int(11) DEFAULT NULL,
  `GOL_1` int(11) DEFAULT NULL,
  `GOL_2` int(11) DEFAULT NULL,
  `JUEGO_1` int(11) DEFAULT '99',
  `JUEGO_2` int(11) DEFAULT '99'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-- Indices de la tabla `QUINIELA_PREDICCIONES`
ALTER TABLE `QUINIELA_PREDICCIONES`
  ADD PRIMARY KEY (`ID`),
  ADD UNIQUE KEY `quina_predictions_id_uindex` (`ID`),
  ADD KEY `quina_predictions_game_id_fk` (`JUEGO`),
  ADD KEY `quina_predictions_quinela_id_fk` (`QUINIELA`),
  ADD KEY `quina_predictions_users_id_fk` (`USUARIO`),
  ADD KEY `quina_predictions_countries_id_fk` (`JUEGO_1`),
  ADD KEY `quina_predictions_countries2__fk` (`JUEGO_2`);
```

-----

```
-- Estructura de tabla para la tabla `QUINIELA_PREMIOS`
DROP TABLE IF EXISTS `QUINIELA_PREMIOS`;
CREATE TABLE `QUINIELA_PREMIOS` (
  `ID` int(11) NOT NULL,
```

```
`QUINIELA` int(11) DEFAULT NULL,  
`PUESTO` int(11) DEFAULT NULL,  
`PREMIO` varchar(120) DEFAULT NULL,  
`DESCRIPCION` varchar(250) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Indices de la tabla `QUINIELA\_PREMIOS`

```
ALTER TABLE `QUINIELA_PREMIOS`  
  ADD PRIMARY KEY (`ID`),  
  ADD UNIQUE KEY `quinaela_award_id_uindex` (`ID`),  
  ADD KEY `quinaela_award_quinela_id_fk` (`QUINIELA`);
```

-----

-- Estructura de tabla para la tabla `QUINIELA\_TIPOS`

```
DROP TABLE IF EXISTS `QUINIELA_TIPOS`;  
CREATE TABLE `QUINIELA_TIPOS` (  
  `ID` int(11) NOT NULL,  
  `NOMBRE` varchar(60) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Indices de la tabla `QUINIELA\_TIPOS`

```
ALTER TABLE `QUINIELA_TIPOS`  
  ADD PRIMARY KEY (`ID`),  
  ADD UNIQUE KEY `quinaela_type_id_uindex` (`ID`);
```

-----

-- Estructura de tabla para la tabla `QUINIELA\_USUARIOS`

```
DROP TABLE IF EXISTS `QUINIELA_USUARIOS`;
```



```
CREATE TABLE `QUINIELA_USUARIOS` (  
  `ID` int(11) NOT NULL,  
  `QUINIELA` int(11) DEFAULT NULL,  
  `USUARIO` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-- Indices de la tabla `QUINIELA_USUARIOS`
```

```
ALTER TABLE `QUINIELA_USUARIOS`  
  ADD PRIMARY KEY (`ID`),  
  ADD UNIQUE KEY `quinaela_users_id_uindex` (`ID`),  
  ADD KEY `quinaela_users_quinela_id_fk` (`QUINIELA`),  
  ADD KEY `quinaela_users_users_id_fk` (`USUARIO`);
```

```
-- -----
```

```
-- Estructura de tabla para la tabla `UBICACIONES`
```

```
DROP TABLE IF EXISTS `UBICACIONES`;  
CREATE TABLE `UBICACIONES` (  
  `ID` int(11) NOT NULL,  
  `NOMBRE` varchar(60) NOT NULL,  
  `CIUIDAD` varchar(50) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-- Indices de la tabla `UBICACIONES`
```

```
ALTER TABLE `UBICACIONES`  
  ADD PRIMARY KEY (`ID`),  
  ADD UNIQUE KEY `locations_id_uindex` (`ID`);
```

-----

-- Estructura de tabla para la tabla `USUARIOS`

DROP TABLE IF EXISTS `USUARIOS`;

CREATE TABLE `USUARIOS` (

`ID` int(11) NOT NULL,

`NOMBRE` varchar(80) DEFAULT NULL,

`CORREO` varchar(80) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

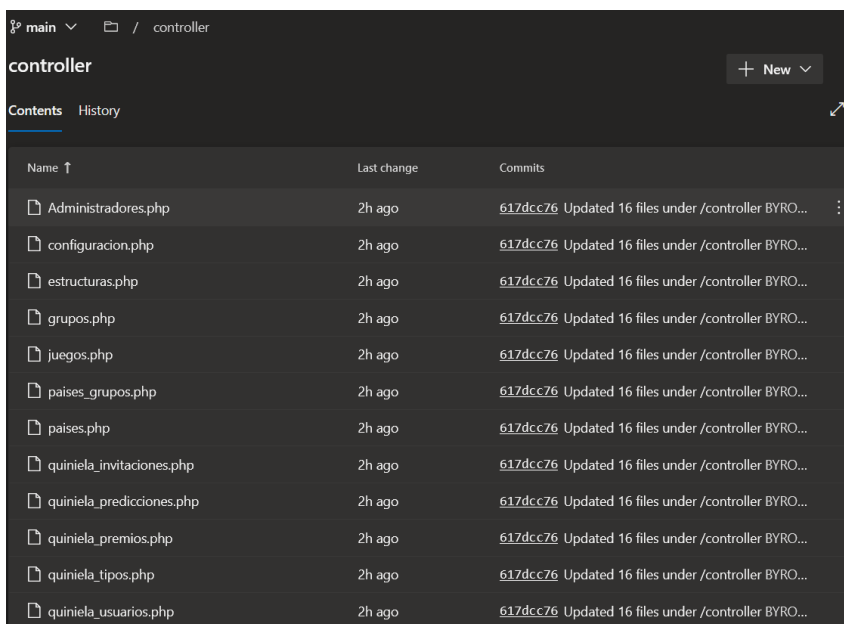
-- Indices de la tabla `USUARIOS`

ALTER TABLE `USUARIOS`

ADD PRIMARY KEY (`ID`),

ADD UNIQUE KEY `users\_id\_uindex` (`ID`);

6. La parte del código se realizó en lenguaje PHP.



Name ↑	Last change	Commits
Administradores.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
configuracion.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
estructuras.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
grupos.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
juegos.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
paises_grupos.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
paises.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
quiniela_invitaciones.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
quiniela_predicciones.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
quiniela_premios.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
quiniela_tipos.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...
quiniela_usuarios.php	2h ago	<a href="#">617dcc76</a> Updated 16 files under /controller BYRO...

```
<?php
header('Content-Type: application/json');

require_once("../config/conexion.php");
require_once("../models/Administradores.php");
$administradores = new Administradores();

$body = json_decode(file_get_contents("php://input"),true);

switch($_GET["op"]){
    case "GetAll":
        $datos=$administradores->get_administradores();
        echo json_encode($datos);

        break;

    case "GetId":
        $datos=$administradores->get_administradores_x_id($body["adm_id"]);
        echo json_encode($datos);

        break;

    case "Insert":
        $datos=$administradores-
>insert_administradores($body["admins_id"],$body["admins_nombre"],$body["admi
ns_correo"],$body["admins_password"],$body["admins_estado"]);
        echo "Se Registro Correctamente";

        break;

    case "Update":
        $datos=$administradores-
>update_administradores($body["admins_id"],$body["admins_nombre"],$body["ad
mins_correo"],$body["admins_password"],$body["admins_estado"]);
        echo "Update Correcto";

        break;

    case "Delete":
        $datos=$administradores->delete_administradores($body["admins_id"]);
        echo json_encode("Delete Correcto");

        break;
```

```
}
```

```
?>
```