

# Quiniela Copa Mundial Qatar 2022

Proyecto final de Bootcamp Ruby on Rails de Código Facilito

Jorge Elliott Ramírez Guardado

Alumno

## DESCRIPCIÓN DEL PROYECTO

En la cultura de los deportes, en particular del fútbol, es común que grupos de personas se organicen para jugar una "Quiniela", que consiste en pronosticar el resultado de los partidos de algún torneo y competir por quién acierta más pronósticos.

Mi proyecto consiste en una aplicación web que permita a los aficionados del fútbol jugar una Quiniela con sus amigos para el torneo "Copa Mundial de la FIFA Qatar 2022".

Funcionalidades:

- La aplicación web permite que cualquier usuario se dé de alta con su nombre, avatar (opcional), correo electrónico y contraseña.
- Los usuarios registrados podrán crear grupos privados.
- El creador de un grupo puede enviar invitaciones por email a sus amigos para unirse al grupo.
- Los usuarios registrados podrán unirse a un grupo privado existente a través de un código de invitación.
- Nadie puede acceder a la información de un grupo al que no ha sido invitado.
- Los miembros del grupo podrán elegir sus pronósticos de los partidos hasta antes de que inicie el partido correspondiente.
- Los usuarios podrán ver puntuación y resultados propios y de los demás miembros del grupo.
- Solo los creadores de grupos pueden eliminar sus grupos o expulsar miembros.

## OBJETIVOS

El objetivo principal del proyecto fue implementar los conocimientos adquiridos a lo largo del Bootcamp.

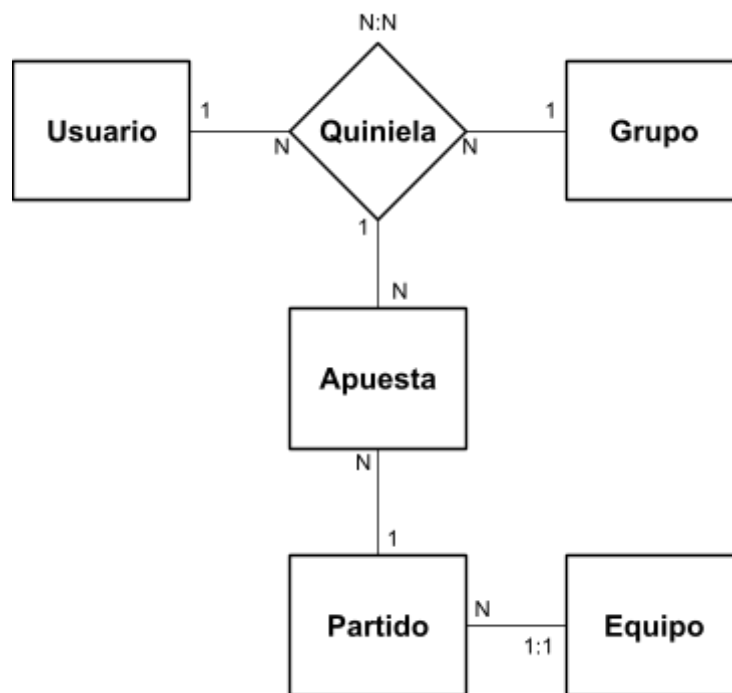
En particular implementar:

- **Modelos** relacionados a través de **Active Model**.
- Almacenamiento de **archivos** a través de **Active Storage**.
- Catálogos **CRUD** a través de peticiones **REST**.

- Formularios a través de **Form Helpers**.
- **Action Mailer** y **Active Job**.
- **Autenticación** con Devise y **autorización** con Pundit.
- Implementación del **framework CSS** Bulma y Sass a través del Asset Pipeline.
- **Despliegue** a un entorno de producción.

## IMPLEMENTACIÓN

- Para satisfacer las necesidades de la aplicación se crearon los siguientes modelos:



- Los modelos se implementaron con los **generadores de Rails** y a partir de ahí se personalizaron los **modelos, controladores y vistas** correspondientes.
- Para configurar **Active Storage** se crearon tablas adicionales para administrar los **avatares** de los usuarios.
- El modelo Usuario fue el que se utilizó para configurar la autenticación a través de **Devise**.
- Para controlar que usuarios sin registrarse no pudieran acceder al sitio así como que un usuario registrado solo pudiera acceder a información correspondiente a sus grupos, se implementó autorización con **Pundit**.
- Los formularios en las vistas se crearon a través de **Action View Form Helpers** para **validar y proteger** peticiones.
- Los parámetros enviados en formularios fueron validados a través de la convención de Rails de **Strong Params**.
- Las invitaciones por email se accionan a través de un formulario que dispara un evento de **Action Mailer** como trabajo en **segundo plano**.

- Para que las vistas fueran visualmente más amigables y atractivas se instaló el **framework CSS Bulma** a través del nuevo **CSS Bundling** implementado en Rails 7. Este se integra al **Asset Pipeline**, evitando usar Webpack.
- La aplicación se desplegó a producción a través de **Heroku**.

## RETOS

Uno de los principales retos de este proyecto fue en un formulario de "Quiniela" poder actualizar sus correspondientes "Apuestas". De igual forma, al crear un grupo nuevo, debía poder en el formulario de "Grupo" crear su "Quiniela" correspondiente junto con sus "Apuestas" de esa "Quiniela" por lo que fue necesario implementar **formularios anidados** de hasta 2 niveles de profundidad.

Esto se implementó respetando las convenciones para formularios de Rails de usar **Form Builders** y **Strong Params**.

## PERSPECTIVAS

Mi propósito es seguir añadiendo funcionalidades a la aplicación, tales como:

- **Actualizaciones en tiempo real** para los partidos en vivo.
- Autenticación con **OAuth**.
- **Pruebas** unitarias y de integración.
- **Validación** y generación de **variantes** para los avatares.