TEC | Tecnológico de Costa Rica

MANUAL TÉCNICO

“INSCRIPCIONES FEDEPAT”

ELABORADO POR

RUY FUENTES GONZÁLEZ
JOSUÉ GERARDO GUTIÉRREZ MORA
ANDREY LOAIZA HERNÁNDEZ
ANDYER ALPIZAR SOLÍS

SAN JOSÉ
2022

RESUMEN

El presente documento es un manual técnico donde se muestra el manejo interno del sistema web en desarrollo para la FEDEPAT, este archivo tiene cómo fin ser una guía para futuros desarrolladores que quieran continuar con el proyecto así cómo también una forma de documentar la tecnología y las multiples funcionalidades que tiene el sistema web de modo que se muestra de una manera más representativa todo el proceso interno que conllevó la creación de la aplicación.

Palabras clave: sistema, FEDEPAT, manual.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Sobre este manual

Este manual tiene cómo fin servir de guía sobre la arquitectura del sistema web además de mostrar con detalle las tecnlogías utilizadas en su creación para preveer inconvenientes de versiones con las herramientas utilizadas. Además, sirve para modular el trabajo de manera que si se desea consultar algún apartado del trabajo se pueda ir directamente a él y ver sus detalles documentados.

1.2. Descripción y Alcance del proyecto

1.2.1. Antecedentes

El proyecto presente es solicitado por la Federación Costarricense de Patinaje y Deporte (FEDEPAT), específicamente por el departamento de administración ya que son ellos quienes necesitan un sistema para poder administrar de manera más eficaz las miles de solicitudes que reciben anualmente por parte de los interesados en subscribirse a los diferentes eventos y actividades que esta organización ofrece.

Cómo se mencionaba anteriormente, este trabajo tiene cómo fin brindar una solución a la necesidad de la FEDEPAT de poder mantener en un sólo lugar todas las solicitudes de subscripción que reciben y ahí mismo poder manipularlas de manera que puedan aceptarlas o recharzarlas según el criterio del administrador. Además de esto, el presente proyecto otorga el manejo de la información de los solicitantes de manera centralizada, es decir, en este mismo sistema web los administradores tendrán la oportunidades de ver la información de los solicitantes de manera que podrán administrarla, filtrarla o usarla para ponerse en contacto con un usuario en cuestión.

Este proyecto planea tener un gran impacto ya que según el propio gestor administrativo esto sería de gran importancia tanto para la beneficio de la FEDEPAT al tener un mejor control de sus usuarios cómo para los propios usuarios que se registran ya que se les haría mucho más simple la subscripción a aquello que ellos desean.

Este trabajo traerá bastantes beneficios para los adminsitradores de la FEDEPAT ya que podrán administrar de una manera centralizada las solicitudes que reciben además de poder unificar la información de los usuarios registrados en un sólo sistema web brindandoles la oportunidad de manejar dicha información y poder tener más control a la hora de realizar sus eventos. Por otro lado, la creación y aviso de eventos será más simple y la comunicación con las personas se volverá más sencilla debido a lo que se mencionaba anteriormente sobre la centralización de la información la cual será más fácil de buscar.

1.2.2. Objetivos

- **Objetivo General**
  - Crear una aplicación web para el registro de usuarios para la FEDEPAT y así los administradores tengan todos los datos en un sólo lugar de manera ordenada permitiendo su administración.
- **Objetivos Específicos**
  - Crear una solución para que los usuarios interesados en los eventos de la FEDEPAT puedan registrarse fácilmente.
  - Unificar la información de los usuarios registrados en la FEDEPAT para que toda esta se almacene en un sólo lugar volviendola más accesible y ordenada.
  - Facilitar la creación e inscripción a eventos que la FEDEPAT realiza.

1.2.3. Contacto

Para consultas y soporte, contacte a los desarrolladores:

Nombre	Teléfono	Correo
Andrey Loaiza	+(506) 84945244	andreyloaizah@estudiantec.cr
Josué Gutiérrez	+(506) 88505223	josuegerardo09@estudiantec.cr
Ruy Fuentes	+(506) 86636749	rafuentesg26@estudiantec.cr
Andyer Alpízar	+(506) 88094673	andyeralpizar24@estudiantec.cr

Cuadro 1.1: Tabla de contactos para soporte

Capítulo 2

Especificación de funcionalidad

2.1. Descripción del producto

En la tercera iteración (MVP) el producto tendrá una página de registro, los usuarios se podrán registrar, se podrán registrar deportistas y a su vez un administrador podrá aceptar o rechar la agregación de deportistas a un equipo al igual que aceptar o rechazar usuarios encargados para acceder al sistema, también será posible agregar dichos deportistas a las competencias que realice la FEDEPAT, las cuales podrán ser creadas en el sistema por un administrador, además, en las competencias se podrán realizar diferentes filtros de información como por ejemplo filtraciones de competencias por tipo y provincia en la que se realizará.

2.1.1. Historias de usuario

Título de la épica

Código	Descripción	Criterios de Calidad
RQ001	Creación de la base de datos	La base se conecta correctamente, no muestra ningún error o mensaje de *warning*
RQ002	Conexión con la base de datos de máquina virtual	No se presentan errores en la conexión, es instantánea y cualquier usuario en otra computadora la puede usar
DC001	Creación de diagramas de clase	Los diagramas están presentes en los documentos con las correciones apuntadas por el supervisor del proyecto
DC002	Creación de diagrama conceptual	El diagrama se encuentra en los documentos oficiales con las correciones apuntadas por el supervisor del proyecto
DC003	Creación de diagrama relacional	El diagrama se encuentra en los documentos oficiales con las correciones apuntadas por el supervisor del proyecto

Código	Descripción	Criterios de Calidad
DC004	Creación de documento patrón arquitectónico	El patrón se encuentra en los documentos oficiales con las correciones apuntadas por el supervisor del proyecto
DC005	Creación del manual de usuario	El manual de usuario cubre y explica **todos** los requerimientos funcionales, además de tomar en cuenta las correciones que el supervisor haga sobre este documento
DC006	Creación de manual técnico	Responde a las preguntas que se encuentran en la plantilla otorgada por el supervisor, además de tomar en cuenta las correciones que el supervisor haga sobre este documento
DC007	Documentación del proyecto	Responde a las preguntas que se encuentran en la plantilla otorgada por el supervisor, además de tomar en cuenta las correciones que el supervisor haga sobre este documento
CL001	El cliente quiere que un admin se puedan registrar	El sistema registra al usuario en la base de datos, se ven sus datos en la misma y corresponden con lo que se insertó
CL002	El cliente quiere que un admin pueda hacer login	El sistema permite a un administrador iniciar sesión sólo si este se encuentra registrado, el propio sistema lo llevará a otra página que indica que ha iniciado sesión
CL003	El cliente quiere que un admin tenga una página de inicio	Cuando un admin registrado hace *login* el sistema lo lleva a una página de inicio que sólo tendrá un mensaje que indica que se encuentra en la página de inicio del administrador

Código	Descripción	Criterios de Calidad
CL004	El cliente quiere que un delegado se puedan registrar	El usuario delegado podrá registrarse en el campo de registro correspondiente con los delegados, al insertar sus datos recibirá una confirmación que lo registrará. Sus datos se podrán ver en la base de datos luego de dicha confirmación
CL005	El cliente quiere que un delegado pueda hacer login	Si un delegado ya se ha registrado este al insertar correctamente sus datos podrá ir a una página de inicio que le indica que ha ingresado correctamente a sus cuenta
CL006	El cliente quiere que un delegado tenga una página de inicio	Cuando el delegado inicia sesión puede ir a una página de inicio que sólo tendrá un mensaje de bienvenida indicando que ha ingresado satisfactoriamente a su cuenta
CL007	El cliente quiere que un administrador puede editar su informacion	Es posible que un administrador edite su propia información, basta con seleccionar la opción y actualzuar un formulario
CL008	El cliente quiere que el delegado puede editar su informacion	Es posible que un delegado edite su propia información, basta con seleccionar la opción y actualzuar un formulario
CL009	El cliente quiere que un administrador puede eliminar a otro administrador	un administrador tiene la posibilidad de eliminar otros administradores a excepción del admnistrador principal del sistema
CL010	El cliente quiere que un administrador puede eliminar a un delegado	Un administrador puede eliminar a cualquier usuario delegado del sistema si lo desea
LO-31	Admin acepta o rechaza nuevos jugadores	Un administrador puede aceptar y rechazar las solicitudes de registros de nuevos jugadores

Código	Descripción	Criterios de Calidad
LO-29	El delegado puede ver la lista de competidores	Los usuarios delegados pueden observar la listas de los competidores pertenecientes de su equipo
LO-27	Registrar competidor	Un delegado puede registrar un competidor al sistema
LO-30	Editar competidor	Un usuario delegado puede editar la informaciónde un competidor
LO-28	Eliminar competidor	Un usuario delegado puede eliminar a un competidor de su equipo en caso de que este ya no pertezca a el
LO-21	Documentación final del sprint	Se cuenta con toda la documentación del sprint
LO-18	Manejo de imágenes para el registro	No es posible el manejo de imagenes en el sistema
LO-23	Creación de términos y condiciones	El sistema cuenta con el apartado de términos y condiciones sin embargo, no se posee la información de los términos que debe brindar la FEDEPAT
LO-29	Validar solicitudes de parte del admin	Ya se encuentra validado todas las solicitudes en la parte del Administrador
LO-24	Actualizar el diseño de la interfaz(pendiente del sprint 1)	El sistema ya cuenta con un diseño de interfaz mucho más atractivo que en el primer Sprint
LO-26	Documentación inicial del sprint	Se finalizó toda la documentación del Sprint
LO-20	Actualizar el diseño de la interfaz(parte del sprint 2)	El sistema ya cuenta con un diseño de interfaz mucho más atractivo que en el primer Sprint
LO-25	Corregir el bug de cerrar sesión	No existe un bug para cuando un usuario cierra sesión
LO-22	Actualizacion de datos requeridos en los registros	Ya se puede actualizar los datos que se estaban requiriendo en los registros
LO-37	Editar competencia(front end)	El sistema con un buen diseño de la interfaz del sistema en el apartado de Editar Competencia

Código	Descripción	Criterios de Calidad
LO-38	Editar competencia(back end)	Es posible que un usuario administrador pueda editar la información de una competencia
LO-39	Eliminar competencia (front end)	El sistema con un buen diseño de la interfaz del sistema en el apartado de Eliminar Competencia
LO-40	Eliminar competencia (back end)	Es posible que un usuario administrador pueda eliminar una competencia del sistema
LO-41	Registrar competidor en una competencia (front end)	El sistema con un buen diseño de la interfaz del sistema en el apartado en donde se puede registrar a un competidor en una competencia
LO-42	Registrar competidor en una competencia (back end)	Un usuario delegado puede registrar a sus competidores a las competencias de la FEDEPAT
LO-43	Eliminar competidor de una competencia(front end)	El sistema con un buen diseño de la interfaz del sistema en el apartado de Eliminar un competidor de una competencia
LO-44	Eliminar competidor de una competencia(back end)	Un usuario puede eliminar a un competidor de una competencia si éste lo desea
LO-45	Terminos y condiciones en una competencia	Las competencias cuentan con un apartado de términos y condiciones, sin embargo la información de los términos aun no ha sido brindada porla FEDEPAT
LO-46	Creacion de filtros para competencias(front end)	El sistema cuenta con la interfaz para la realización de filtros de las competencias
LO-47	Creacion de filtros de competencia (back end)	Un usuario puede realizar filtraciones en las competencias, por ejemplo, es posible filtrar una competencia por tipo y provincia
LO-48	Manejo de correos	El sistema no cuenta con el manejo de correos

Código	Descripción	Criterios de Calidad
LO-49	Documentacion de incio de sprint	Se cuenta con la documentación del inicio del sprint
LO-50	Documentacion final de sprint	Se finalizó toda la documentación Final del proyecto
LO-51	Crear competencia(back end)	El sistema cuenta con un buen diseño de la interfaz del sistema en el apartado de Crear Competencia
LO-52	Crear competencia(front end)	Es posible que un usuario administrador pueda crear una competencia en el sistema para que los delegados puedan inscribir a sus competidores
LO-55	Experiencia de usuario	La experiencia del usuario es sastifactoria, y el sistema cuenta con la mayoría de las necesidades del usuario

Capítulo 3

Arquitectura del sistema

3.1. Diseño general del sistema

Se puede observar la arquitectura del sistema. Se utilizara la arquitectura Modelo, Vista, Controlador(MVC), debido a que el caso que se está trabajando es un sistema de inscripciones por medio de usuarios, y utilizar MVC para este caso se considera la mejor opción.

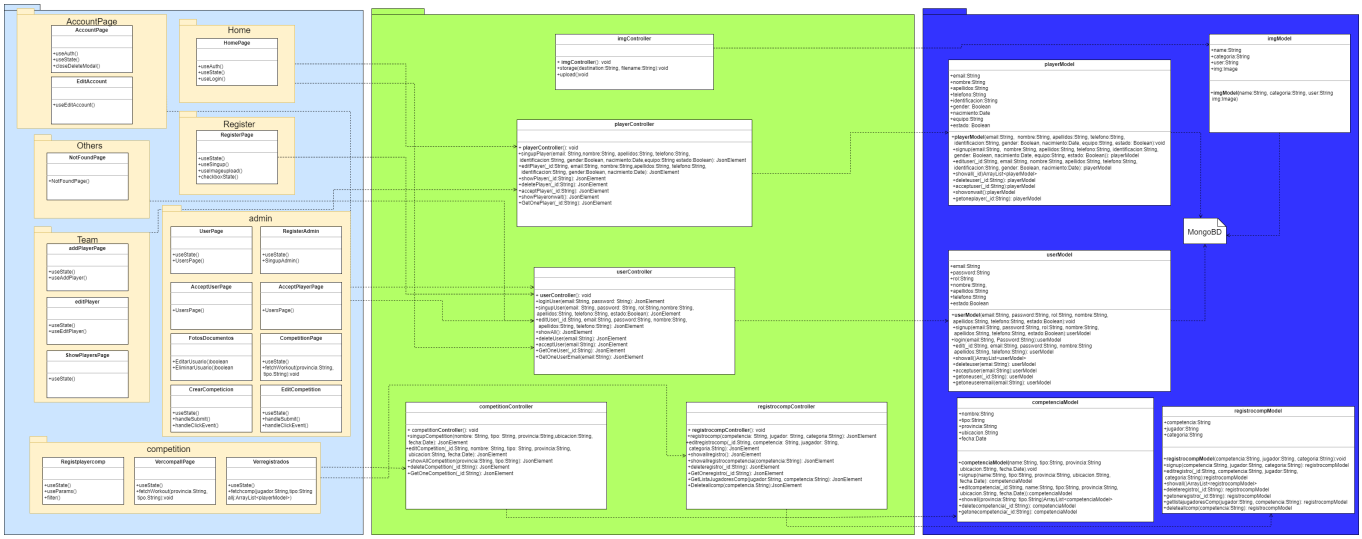

Figura 3.1: Diseño arquitectónico

Para fines del tercer Sprint en el paquete de vista se cuenta con diferentes paquetes con vistas que requiere el sistema, las cuales son:

- **Account Page:** Esta vista muestra los datos de la cuenta del usuario que se encuentra en el sistema, también contiene, el apartado para que el usuario pueda editar sus datos.

- **Home:** Esta vista mostrará el menu principal del sistema con las herramientas necesarias, segun el tipo de usuario que haya ingresado (Encargado de un equipo o administrador)

- **Team:** Este paquete de vistas contiene las vistas necesarias que tienen que ver con la muestra de deportistas, en donde el encargado del equipo puede visualizarlos y además editar o actualizar los datos de los miembos, si lo desea.

- **admin:** El paquete admin, hace referencia a todas las vistas que necesitará el usuario administrador, entre ellas está, el registro de nuevos administradores, las solicitudes que llegan por parte de los encargados de los equipos ya sea para ingresar al sistema o para permitir que se agreguen nuevos competidores al sistema, entre otras funcionalidades que necesitan los administradores.

- **Register:** Este paquete tiene la vista que permite registrar los datos de un nuevo usuario, para que este pueda acceder al sistema.

- **Others:** El paquete de Others contiene vistas auxiliares que puede necesitar el sistema, por el momento se cuenta con una página de tipo No Found Page, Error 404 en caso de que ocurra un problema con el sistema.

- **competition:** Este paquete cuenta con las vistas en donde ya un administrador puede crear competencias, para que los delegados puedan inscribir a sus atletas, las vistas permiten a los usuarios poder visualizar dichas competencias y realizar diferentes filtraciones de las mismas.

El Paquete del Controlador se encarga de solicitar o dar instrucciones del sistema entre las vistas y los modelos, en este sprint contamos con tres controladores uno por cada modelo:

- **imgController:** Este controlador se encarga de realizar el manejo de archivos como imagenes, para recolectar los datos necesarios que necesita la clase ImgModel, y de esta manera poder guardalas en la base de datos.

- **userController:** Esta clase realizará todas las transacciones entre las vistas y los modelos que tengan que ver con los usuarios, también para obtener la información almacenada en la base de datos de Mongo, funcionalidades como la de aceptar o rechazar las peticiones de los encargados del equipo y demás funciones administrativas es para lo que se usa este controlador.

- **playerController:** Básicamente es el controlador encargado de todas las transacciones que tienen que ver los deportistas, ya sea para mostraslos o aceptarlos y rechazarlos en el sistema.

- **competitionController:** Es utilizado para las transacciones con el modelo competenciaModel, el cual posee los datos de una competencia, como el nombre, tipo, ubicación, entre otros, de igual manera puede acceder a los métodos de la misma encargados de la edición, eliminación y demás funcionalidades que poseen las competencias.

- **registrocompController:** El controlador se encarga de la transferencia de datos con el modelo registrocompModel, el cual es utilizado para poder registrar a los jugadores en las competencias de acuerdo a la categoría en la que el delegado lo inscriba .

En el paquete de Modelo se tiene el manejo de la persistencia de datos e instrucciones para poder hacer las transferencias, este está compuesto por las siguientes clases:

- **ImgModel:** Esta clase recolecta los datos necesarios de una imagen para poder ser guardada en la base de datos.

- **playerModel:** Realiza las transacciones que el Controlador solicita con respecto a los depostistas, además de realizar las funciones que tengan que ver con visualizar o agregar deportistas a un equipo.

- **userModel:**Realiza las transacciones que el Controlador solicita con respecto a los usuarios administradores y encargados ,permite controlar toda función que tenga que ver con el manejo de los datos de un usuario, tanto como sus registros y visualización de datos personales.

- **competenciaModel:** Está clase se comunica con el controlador competitionController, para la creación, edición, eliminación entre otras funciones ameritan las competencias, además también es capaz de realizar filtraciones de competencias, utilizando la provincia y el tipo.

- **registrocompModel:** Este modelo realiza las transacciones con registrocompController, para funcionalidades de registros con las competencias. Es la que permite que los delegados puedan registras a sus competidores a las competencias y los administradores puedan visualizar a los participantes de estas.

3.1.1. Diseño de la persistencia

El diseño de persistencia fue creado a partir de la idea de una página de login tanto para un administrador cómo para un delegado, de ahí se extiende lo que cada uno de estos necesita.

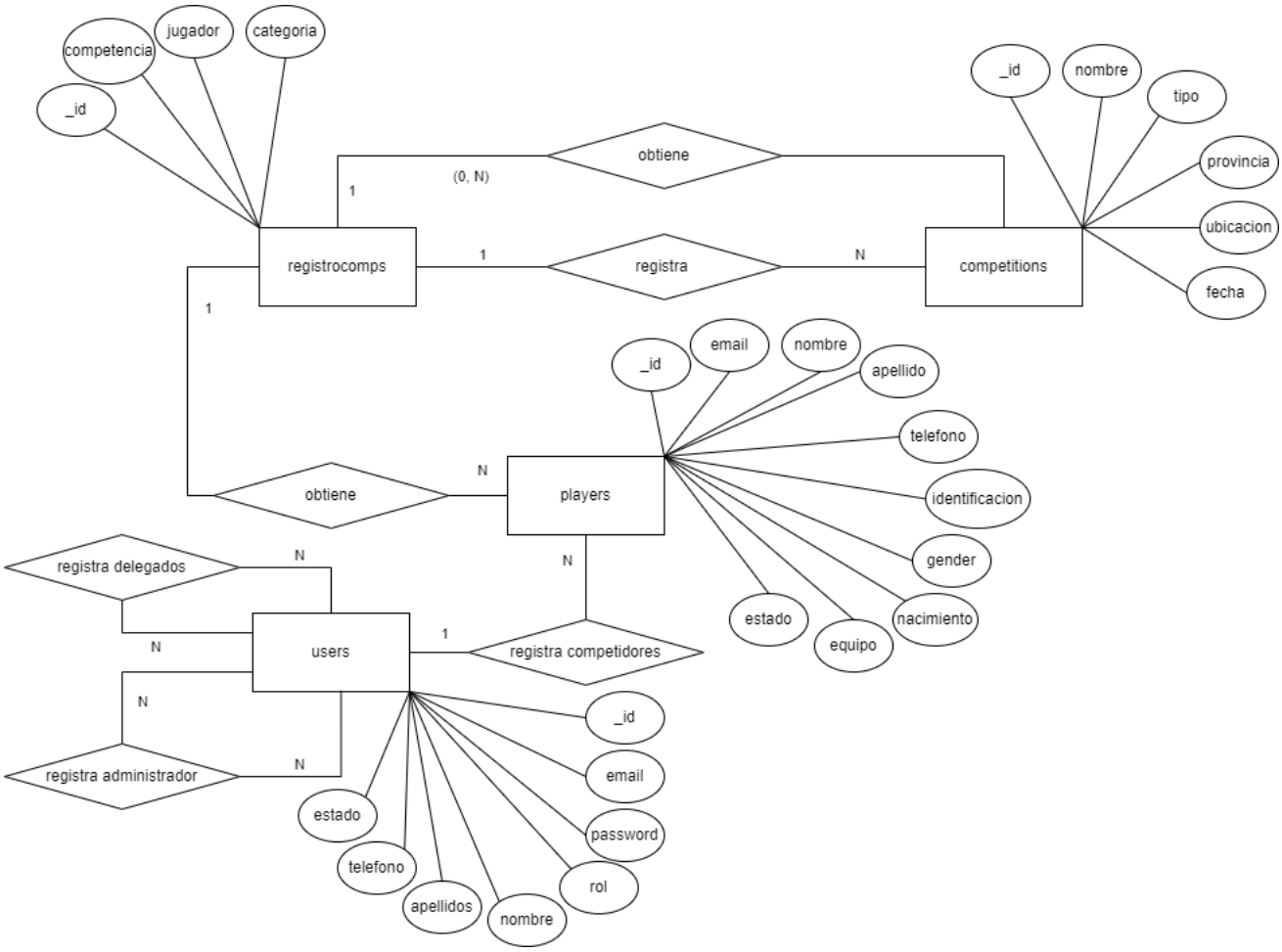

Figura 3.2: Diseño de entidad-relación

Diagrama relacional

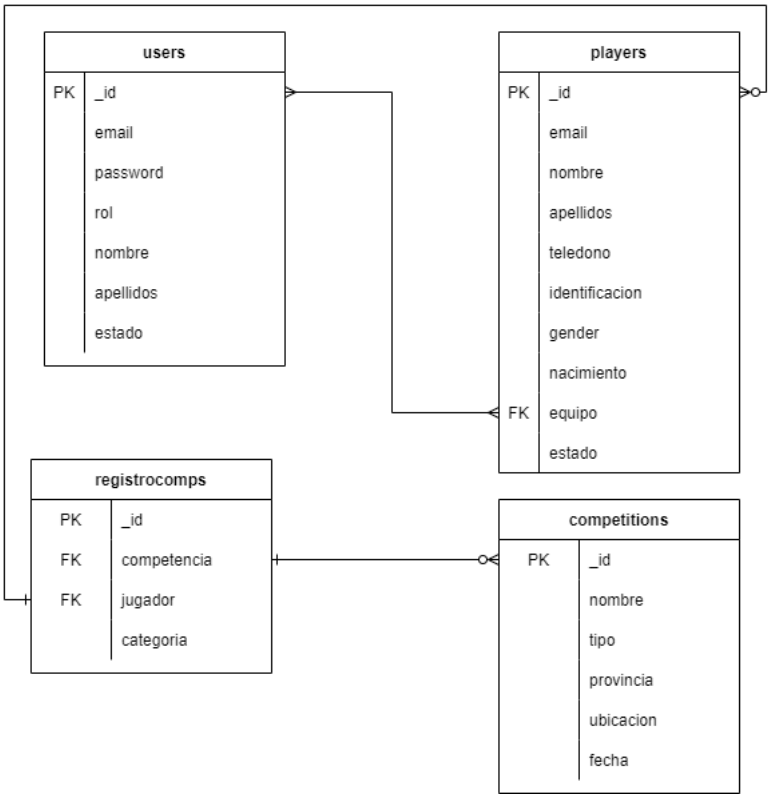

Figura 3.3: Diagrama relacional

Debido a las similitudes que presenta la información del administrador con la del delegado, se utilizara la misma tabla para ambos y se identificara con una variable booleana que tipo de usuario se ingreso en la base de datos.La tabla de registrocomps necesita los id de los players y de las competitions para realizar la asociación de que competidores están registrados en cuales competencias.

Cambios

- Para este tercer sprint si se registra cambios los cuales son la agregacion de dos nuevas entidades la cuales son la de registrocomps y competitions, el diseño de la arquitectura y diagrama de clases, fue modificado, ahora el sistema cunta con 5 modelos y un controlador para cada uno que se encarga de las transferencias entre las vistas.
- Otro de los cambios fue la restructuración de los diagramas de Entidad-Relacion y Relacional, ésto incluye la agregación de las dos nuevas entidades ya mencionadas que se encargan del registro de competencias y la inscripcion de los competidores a las mismas.

3.1.2. Diagrama de clases

A continuación se muestra el diagrama de clases donde se podrá observar cómo está estructurada la planeación del proyecto y cómo este puede ser escalable

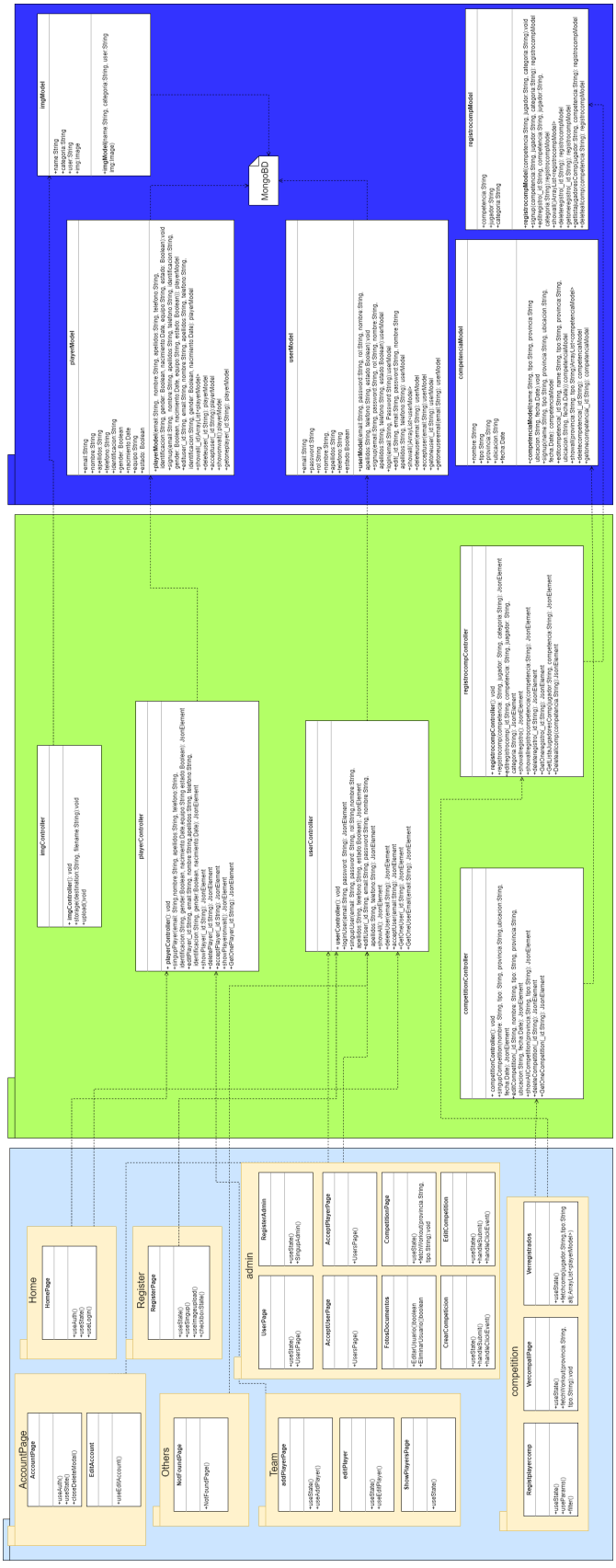

Figura 3.4: IMAGE CAPTION

Capítulo 4

Instrucciones de Instalación

4.1. Entorno de Desarrollo

4.1.1. Dependencias

El trabajo en cuestión fue realizado utlizando el *stack* MERN (Mongo, Express, React, Node) para abarcar todos los apartados necesarios de desarrollo (base de datos, frontend y backend). Para esto es necesario tener las herramientas que el *stack* pide para poder ejecutarse correctamente.

También se debe decir que el desarrollo fue hecho mediante una máquina virtual que corre sobre un sistema operativo Linux CentOS por lo tanto, todas las explicaciones aquí realizadas serán orientadas a dicho sistema operativo.

A continuación se muestran todas las dependencias necesarias para el entorno de desarrollo:

- **NodeJS versión 16:** Para instalar NodeJS simplemente siga esta línea de pasos de instalación por línea de comandos:
  - *curl -fsSL https://rpm.nodesource.com/setup*$_1$*6.x*|*sudobash*−*Estoesparadescargarlaversión*16*deNo*
    sudo yum install -y nodejsEstoesparainstalarlaversióndeNodeinstaladaenelpuntoanterior.
  - *sudo yum install gcc-c++ make* Para instalar las herramientas de desarrollo que son necesarias a nivel interno de la aplicación.
- **ReactJS versión 8:** Necesariamente se debe de tener primero instalado el NodeJS que se explicó en el punto anterior, luego basta con correr el siguiente comando en la línea de comandos:
  - npm install -g create-react-app
- **Express versión 4:** Igualmente se necesita tener primero instalado NodeJS para la instalación de Expresss, de ese modo sólo basta con correr el siguiente comando:
  - npm install express
- **MongoDB versión 6:** Para la instalación de MongoDB se deben seguir las siguientes líneas de comandos:
  - *vim /etc/yum.repos.d/mongodb-org-6.0.repo* Esto es para crear en esa dirección un archivo mediante el editor VIM.

```
[mongodb-org-6.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/6.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-6.0.asc
```

Copie y pegue todo este código en el editor que abrió anteriormente, luego presione la tecla *ESC* y escriba *:wq* para guardarlo y salir.

- *sudo yum install -y mongodb-org* Este comando es para correr el archivo que se creó anteriormente y así poder instalar MongoDB más fácilmente.

- *sudo yum install git* Esto para la instalación de Git py poder clonar fácilmente el repositorio del trabajo.
- Por último, se deben de instalar las siguientes dependencias para que el proyecto creado funcione correctamente:
  - *npm install mongoose* (se debe utilizar la versión 6)
  - *npm install -g nodemon* (se debe utilizar la versión 2)

4.1.2. Código fuente

- Utilizando *GIT* situese en una carpeta y mediante la línea de comandos utilice el código *git clone https://github.com/josuegerardo96/FEDEPAT.git* el cual se va a encargar de descargar todo el código del proyecto, inicialmente se verán únicamente 2 carpetas: *server* y *client* que corresponden a los apartados de backend y frontend.
- Para poder ejecutar el código fuente del backend basta con ubicarse en la carpeta de *server* y en el terminal escribir el comando *npm install* para que se instalen las dependencias necesarias y propias del proyecto
- Para poder ejecutar el código fuente del frontend basta con ubicarse en la carpeta de *client* y en el terminal escribir el comando *npm install* para que se instalen las dependencias necesarias y propias del proyecto,

4.1.3. Despliegue en entorno de pruebas

- Luego de la descarga e instalación de los elementos explicados en el apartado de *Código fuente* simplemente basta en primer lugar irse a la carpeta *server* y correr el comando *npm start*. En este punto ya el apartado backend y las conexiones con la base de datos estarían funcionando en tiempo real.

- lLuego de la descarga e instalación de los elementos explicados en el apartado de *Código fuente* y haber corrido el backend en el punto anterior, basta con irse a la carpeta de *client* y correr en la terminal el comando *npm start*. En este punto ya el apartado frontend se podrá conectar con los servicios del backend y así desplegar el trabajo realizado en un navegador.

4.2. Entorno de Producción

4.2.1. Dependencias

Si ya se ha corrido con todo lo necesario para el entorno de desarrollo entonces para el entorno de producción no queda más que instalar las siguientes dependencias para el correcto funcionamiento del sistema web.

- *npm install bcrypt* para el apartado de encriptar las contraseñas.
- *npm install validator* para poder manejar las validaciones de contraseñas y correos escritos de manera correcta.
- **npm install jsonwebtoken** para el fácil manejo de archivos JSON.
- **npm install moment** para mostrar formatos de fecha.
- **npm install react-icons –save** para hacer uso de iconos especiales.

4.2.2. Instrucciones de instalación

Una vez instalado todo lo necesario y establecido en el apartado de entorno de desarrollo y de producción, el siguiente paso depende de la finalización del código fuente, al finalizarse y procurar que se encuentre libre de errores, se debe realizar un build en la carpeta raíz y aplicar el comando ”npm build”:

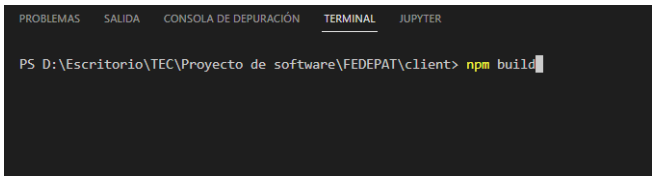

Figura 4.1: Diseño arquitectónico

Al ejecutarse el comando se creará una carpeta automaticamente, la cual corresponde al apartado del cliente ya procesado, en caso de realizar actualizaciones o cambios en el código, se deberá repetir el proceso de la creación del build.

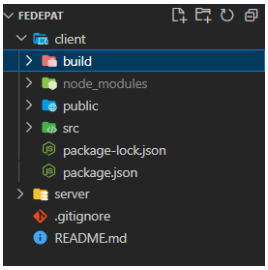

Figura 4.2: Diseño arquitectónico

Seguidamente se debe copiar todo el contenido del build y lo agregamos a una carpeta llamada public en el backend el cual se encuentra en la carpeta server, cabe recalcar que el sistema es ejecutado en el mismo puerto del server, para este caso el utilizado es el puerto 4000.

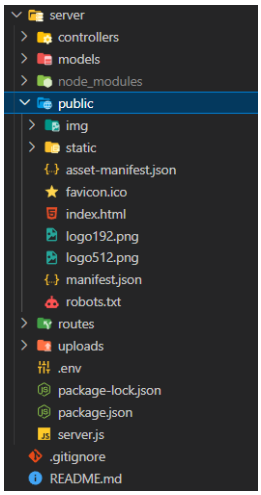

Figura 4.3: Diseño arquitectónico

Finalmente, cuando ya se tiene el contenido en la carpeta ya mencionada, utilizando express se debe crear la ruta que inicia el build ya procesado, ésto se puede realizar de la siguiente forma:

```
1  // go to react and start!
2  app.use(express.static('public'))
3  app.get('*' , (req, res)=>{
4      res.sendFile(path.join(__dirname, '/public/index.html'));
5  })
6
```

Figura 4.4: Diseño arquitectónico

4.2.3. Acceso al Sistema

- El siguiente link le permitirá accesar al sistema: http://144.22.40.150:4000/

- Las credenciales principales del Administrador son:

  - **Usuario:** fedepat123@gmail.com

  - **Contraseña:** Contraseña: Fedepat123.

listings

Capítulo 5

Métodos del API

En esta sección detalla los métodos que se tienen publicados en la sección back-end. Cada método incluye la ruta, los datos respectivos que se envían, al igual que su respuesta.

API: Post: localhost:4000/api/user/login
Request:

```
{
    "email":"loaizah@gmail.com",
    "password":"ABCabc123!"
}
```

Response:

```
{
    "email": "loaizah@gmail.com",
    "nombre": "Minor",
    "apellidos": "Loaiza",
    "telefono": "60391147",
    "user": {
        "_id": "6333b46109120c5f22ddf4d3",
        "email": "loaizah@gmail.com",
        "password": "$2b$10$9/eFnr.fSWQNQdqkTLNGHO1Rlmy5M0JpHjpVrPJCfe7K
        ztUOldoyy",
        "rol": "delegado",
        "nombre": "Minor",
        "apellidos": "Loaiza",
        "telefono": "60391147",
        "estado": true,
        "__v": 0
    }
}
```

API: Post: localhost:4000/api/user/signup
Request:

```
{
    "email":"andreyloaizah1234@gmail.com",
    "password":"TecSanjose2022@",
    "rol":"admin",
    "nombre":"Andrey",
```

```
    "apellidos":"Loaiza",
    "telefono":"224222",
    "estado":0
}
```


Response:


```
{
    "email": "andreyloaizah1234@gmail.com",
    "user": {
        "email": "andreyloaizah1234@gmail.com",
        "password": "$2b$10$XqAc0lw0XQfMfnSWwq1L3.d2HJsoSvKQnFdWxQCQg21Ci0v
        "rol": "admin",
        "nombre": "Andrey",
        "apellidos": "Loaiza",
        "telefono": "224222",
        "estado": false,
        "_id": "6351ad5b1256d941f2403b5d",
        "__v": 0
    },
    "nombre": "Andrey",
    "apellidos": "Loaiza",
    "telefono": "224222"
}
```


API: Post: localhost:4000/api/user/editUser

Request:


```
{
        "email": "andreyloaizah1234@gmail.com",
        "rol": "admin",
        "nombre": "Andrew",
        "apellidos": "Loaiza",
        "telefono": "224222",
        "estado": false,
        "_id": "6351ad5b1256d941f2403b5d",
        "password": "TecSanjose2022@"

}
```


Response:


```
{
    "email": "andreyloaizah1234@gmail.com",
    "user": {
        "_id": "6351ad5b1256d941f2403b5d",
        "email": "andreyloaizah1234@gmail.com",
        "password": "$2b$10$XqAc0lw0XQfMfnSWwq1L3.d2HJsoSvKQnFdWxQCQg21Ci0v
        "rol": "admin",
        "nombre": "Andrey",
        "apellidos": "Loaiza",
        "telefono": "224222",
        "estado": false,
        "__v": 0
},
```

API: GET: localhost:4000/api/user/getusers

Response:


```
{
    "_id": "632b7a2690355d418e61d050",
    "email": "andreyloaizah@gmail.com",
    "password": "$2b$10$dfrB29VpXKsabp1zvpq.QuwdhZpPlJ8GCKaN7eLkIFeEAlXyCnk
    "rol": "admin",
    "nombre": "Andrey",
    "apellidos": "Loaiza Hern ndez",
    "telefono": "84945244",
    "__v": 0,
    "estado": true
},
```


API: DELETE: localhost:4000/api/user/deleteuser

Request:


```
{
"email":"hola@gmail.com"
}
```


Response:


```
{
    "email": "hola@gmail.com",
    "user": {
        "_id": "63518153c6d4196ffba36d4c",
        "email": "hola@gmail.com",
        "password": "$2b$10$1AF5xApiJi8xPEPwFr6zf.6Zahqm3bKeKmzHa4/sZ/RgvWv
        "rol": "delegado",
        "nombre": "Hola",
        "apellidos": "Mundo",
        "telefono": "77894571",
        "estado": false,
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/user/acceptuser

Request:


```
{
    "email":"andreyloaizah1234@gmail.com"
}
```


Response:


```
{
    "email": "andreyloaizah1234@gmail.com",
    "user": {
        "_id": "6351ad5b1256d941f2403b5d",
        "email": "andreyloaizah1234@gmail.com",
        "password": "$2b$10$sg/KofasLt9rCAyAX.ADq.4lC38pnVPK3oYgk/PYAKTw/2C
        "rol": "admin",
        "nombre": "Andrew",
        "apellidos": "Loaiza",
```

```
        "telefono": "224222",
        "estado": false,
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/user/getOneuser

Request:


```
{
    "_id": "6351ad5b1256d941f2403b5d"
}
```


Response:


```
{
    "user": {
        "_id": "6351ad5b1256d941f2403b5d",
        "email": "andreyloaizah1234@gmail.com",
        "password": "$2b$10$sg/KofasLt9rCAyAX.ADq.4lC38pnVPK3oYgk/PYAKTw/2C
        "rol": "admin",
        "nombre": "Andrew",
        "apellidos": "Loaiza",
        "telefono": "224222",
        "estado": true,
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/user/getOneuseremail

Request:


```
{
    "email": "andreyloaizah1234@gmail.com"
}
```


Response:


```
{
    "user": {
        "_id": "6351ad5b1256d941f2403b5d",
        "email": "andreyloaizah1234@gmail.com",
        "password": "$2b$10$sg/KofasLt9rCAyAX.ADq.4lC38pnVPK3oYgk/PYAKTw/2C
        "rol": "admin",
        "nombre": "Andrew",
        "apellidos": "Loaiza",
        "telefono": "224222",
        "estado": true,
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/player/playersignup

Request:


```
{
    "email":"roberto@gmail.com",
```

```
    "nombre":"Roberto",
    "apellidos":"Garcia",
    "telefono":"224222",
    "identificacion":"116910169",
    "gender": 0,
    "nacimiento": "1997-10-26",
    "equipo":"632b7a2690355d418e61d050",
    "estado":0
}
```


Response:


```
{
    "email": "roberto@gmail.com",
    "player": {
        "email": "roberto@gmail.com",
        "nombre": "Roberto",
        "apellidos": "Garcia",
        "telefono": "224222",
        "identificacion": "116910169",
        "gender": false,
        "nacimiento": "1997-10-26T00:00:00.000Z",
        "equipo": "632b7a2690355d418e61d050",
        "estado": false,
        "_id": "6351b1a91256d941f2403b66",
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/player/showplayer

Request:


```
{
    "_id":"634f1b3941b175eab52f00a6"
}
```


Response:


```
[
    {
        "_id": "634f1c2a41b175eab52f00be",
        "email": "Ruys123@gmail.com",
        "nombre": "Ruy",
        "apellidos": "Fuentes",
        "telefono": "84945244",
        "identificacion": "116910169",
        "gender": false,
        "nacimiento": "2004-02-18T00:00:00.000Z",
        "equipo": "634f1b3941b175eab52f00a6",
        "estado": true,
        "__v": 0
    }
]
```


API: DELETE: localhost:4000/api/player/deletePlayer

Request:

```
{
    "_id":"6351b1a91256d941f2403b66"
}
```

Response:


```
{
    "player": {
        "_id": "6351b1a91256d941f2403b66",
        "email": "roberto@gmail.com",
        "nombre": "Roberto",
        "apellidos": "Garcia",
        "telefono": "224222",
        "identificacion": "116910169",
        "gender": false,
        "nacimiento": "1997-10-26T00:00:00.000Z",
        "equipo": "632b7a2690355d418e61d050",
        "estado": false,
        "__v": 0
    }
}
```


API: GET: localhost:4000/api/player/showplayeronwait

Response:


```
[
    {
        "_id": "634e04b5257cc9daadb685df",
        "email": "hola@gmail.com",
        "nombre": "hola",
        "apellidos": "soy",
        "telefono": "222 222 222",
        "identificacion": "116910169",
        "gender": false,
        "nacimiento": "2010-01-01T00:00:00.000Z",
        "equipo": "6333b46109120c5f22ddf4d3",
        "estado": false,
        "__v": 0
    }
]
```


API: Post: localhost:4000/api/player/acceptPlayer

Request:

```
{
    "_id":"634e04b5257cc9daadb685df"
}
```

Response:


```
{
    "player": {
        "_id": "634e04b5257cc9daadb685df",
        "email": "hola@gmail.com",
        "nombre": "hola",
```

```
        "apellidos": "soy",
        "telefono": "222 222 222",
        "identificacion": "116910169",
        "gender": false,
        "nacimiento": "2010-01-01T00:00:00.000Z",
        "equipo": "6333b46109120c5f22ddf4d3",
        "estado": false,
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/player/editplayer

Request:


```
{
    "_id":"6343a6cf6deda3a28b4af6db",
    "email":"Ana12@gmail.com",
    "nombre":"Ana",
    "apellidos":"Alpizar",
    "telefono":"224222",
    "identificacion":"116910169",
    "gender": 1,
    "nacimiento": "1997-10-26"
}
```


Response:


```
{
    "player": {
        "_id": "6343a6cf6deda3a28b4af6db",
        "email": "Ana@gmail.com",
        "nombre": "Ana",
        "apellidos": "Alpizar",
        "telefono": "224222",
        "identificacion": "116910169",
        "gender": true,
        "nacimiento": "1997-10-26T00:00:00.000Z",
        "equipo": "6333b46109120c5f22ddf4d3",
        "estado": true,
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/player/getoneplayer

Request:


```
{
    "_id":"63420427b47797d72d2fe830"
}
```


Response:


```
{
    "player": {
        "_id": "63420427b47797d72d2fe830",
        "email": "Laras12@gmail.com",
```

```
        "nombre": "Lara",
        "apellidos": "Croft",
        "telefono": "85898944",
        "identificacion": "116910169",
        "gender": true,
        "nacimiento": "1984-01-25T00:00:00.000Z",
        "equipo": "6333b46109120c5f22ddf4d3",
        "estado": true,
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/img/imgsave

Request:

```
form Data
const formData = new FormData()
formData.append('name', cedulaphoto.name);
formData.append('fedeimage', cedulaphoto);
formData.append('categoria', 'cedula');
formData.append('user', json.user._id);
```

Response:


```
{
    "saveImage": {
        "name": "image9",
        "img": {
            "data": {
                "type": "Buffer",
                "data": [
                    255,
                    216,
                    255,
                    224,
                    0,
                    .
                    .
                    .
                    .
                  172,
                    202,
                    63,
                    255,
                    217
                ]
            },
            "contentType": "image/png"
        },
        "_id": "6351b4b91256d941f2403b6f"
    }
}
```


API: localhost:4000/api/img/imageget

Request:

```
{
    "_id": "63518153c6d4196ffba36d4c"
}
```


Response:


```
[
    {
        "img": {
            "data": {
                "type": "Buffer",
                "data": [
                    255,
                    216,
                    255,
                    224,
                    0,
                    16,
                    74,
                    70,
                    73,
                    70,
                    .
                    .
                    .
                    .
                    82,
                    166,
                    12,
                    255,
                    217
                ]
            },
            "contentType": "image/png"
        },
        "_id": "63518154c6d4196ffba36d4f",
        "name": "cedula ejemplo.jfif",
        "categoria": "cedula",
        "user": "63518153c6d4196ffba36d4c",
        "__v": 0
    }
]
```


API: Post: localhost:4000/api/competition/singupCompetition

Request:


```
{
    "nombre": "ni os 3",
    "tipo" : "recreativo",
    "provincia":"San Jose",
    "ubicaci n" : "Sabana",
    "fecha":"2022-12-14"

}
```

Response:


```
{
    "nombre": "ni os 3",
    "tipo": "recreativo",
    "provincia": "San Jose",
    "ubicaci n": "Sabana",
    "fecha": "2022-12-14",
    "competition": {
        "nombre": "ni os 3",
        "tipo": "recreativo",
        "provincia": "San Jose",
        "ubicaci n": "Sabana",
        "fecha": "2022-12-14T00:00:00.000Z",
        "_id": "637295b0a59ef0b6a80952b4",
        "__v": 0
    }
}
```


API: Post: localhost:4000/api/competition/editCompetition

Request:


```
{
    "_id":"637295b0a59ef0b6a80952b4",
    "nombre": "ni os Y adultos",
    "tipo" : "recreativo",
    "provincia":"San Jose",
    "ubicaci n" : "Sabana",
    "fecha":"2022-12-14"

}
```


Response:


```
{
    "nombre": "ni os Y adultos",
    "tipo": "recreativo",
    "provincia": "San Jose",
    "ubicaci n": "Sabana",
    "fecha": "2022-12-14",
    "competition": {
        "_id": "637295b0a59ef0b6a80952b4",
        "nombre": "ni os 3",
        "tipo": "recreativo",
        "provincia": "San Jose",
        "ubicaci n": "Sabana",
        "fecha": "2022-12-14T00:00:00.000Z",
        "__v": 0
    }
}
```


API: POST: localhost:4000/api/competition/showAllCompetition

Request:


```
{
    "provincia":"todo",
```

```
    "tipo":"Semiprofesional bota alta en l nea"
}
```


Response:


```
[
    {
        "_id": "636d34bb032a4eff7077c70b",
        "nombre": "competicion cartago",
        "tipo": "Semiprofesional bota alta en l nea",
        "provincia": "Cartago",
        "ubicaci n": "Cartego centro",
        "fecha": "2022-11-10T00:00:00.000Z",
        "__v": 0
    },
    {
        "_id": "636d62885f6cdab831cd97e5",
        "nombre": "La Guanacatecas",
        "tipo": "Semiprofesional bota alta en l nea",
        "provincia": "Guanacaste",
        "ubicaci n": "EL arbol de Guanacaste",
        "fecha": "2022-11-10T00:00:00.000Z",
        "__v": 0
    }
]
```


API: DELETE: localhost:4000/api/competition/deleteCompetition

Request:


```
{
    "_id": "637295b0a59ef0b6a80952b4"
}
```


Response:


```
{
    "_id": "637295b0a59ef0b6a80952b4",
    "competition": {
        "_id": "637295b0a59ef0b6a80952b4",
        "nombre": "ni os Y adultos",
        "tipo": "recreativo",
        "provincia": "San Jose",
        "ubicaci n": "Sabana",
        "fecha": "2022-12-14T00:00:00.000Z",
        "__v": 0
    }
}
```


API: POST: localhost:4000/api/competition/GetOneCompetition

Request:


```
{
    "_id":"636d4934edc80c628711dcf1"
}
```


Response:

```
{
    "competition": {
        "_id": "636d4934edc80c628711dcf1",
        "nombre": "Pat loco 202",
        "tipo": "Pat n recreativo en l nea o tradicional (4 ruedas)",
        "provincia": "Puntarenas",
        "ubicaci n": "Playa jaco",
        "fecha": "2022-11-10T00:00:00.000Z",
        "__v": 0
    }
}
```


API: POST: localhost:4000/api/registcomp/registrocomp

Request:


```
{
    "competencia":"636d62885f6cdab831cd97e5",
    "jugador":"636d62885f6cdab831cd97e5",
    "categoria":"joven"
}
```


Response:


```
{
    "competencia": "636d62885f6cdab831cd97e5",
    "jugador": "636d62885f6cdab831cd97e5",
    "categoria": "joven",
    "competition": {
        "competencia": "636d62885f6cdab831cd97e5",
        "jugador": "636d62885f6cdab831cd97e5",
        "categoria": "joven",
        "_id": "6376efbd8b99dd950885ed27",
        "__v": 0
    }
}
```


API: POST: localhost:4000/api/registcomp/editCompetition

Request:


```
{
    "_id":"6376efbd8b99dd950885ed27",
    "competencia":"636d62885f6cdab831cd97e5",
    "jugador":"636d62885f6cdab831cd97e5",
    "categoria":"adulto"
}
```


Response:


```
{
    "competencia": "636d62885f6cdab831cd97e5",
    "jugador": "636d62885f6cdab831cd97e5",
    "categoria": "adulto",
    "competition": {
        "_id": "6376efbd8b99dd950885ed27",
        "competencia": "636d62885f6cdab831cd97e5",
```

```
        "jugador": "636d62885f6cdab831cd97e5",
        "categoria": "joven",
        "__v": 0
    }
}
```


API: GET: localhost:4000/api/registcomp/showallregristro

Request:

NULL

Response:


```
[
    {
        "_id": "636d6c33170365890c213f0e",
        "competencia": "636d34bb032a4eff7077c70b",
        "jugador": "63420427b47797d72d2fe830",
        "categoria": "Open (15 y m s a os)",
        "__v": 0
    },
    {
        "_id": "636d6c6b170365890c213f32",
        "competencia": "636d34bb032a4eff7077c70b",
        "jugador": "634e04b5257cc9daadb685df",
        "categoria": "Infantil A1 (12 a os)",
        "__v": 0
    },
    {
        "_id": "6376efbd8b99dd950885ed27",
        "competencia": "636d62885f6cdab831cd97e5",
        "jugador": "636d62885f6cdab831cd97e5",
        "categoria": "adulto",
        "__v": 0
    }
]
```


API: DELETE: localhost:4000/api/registcomp/deleteregistro

Request:


```
{
    "_id":"6376efbd8b99dd950885ed27"
}
```


Response:


```
{
    "_id": "6376efbd8b99dd950885ed27",
    "competition": {
        "_id": "6376efbd8b99dd950885ed27",
        "competencia": "636d62885f6cdab831cd97e5",
        "jugador": "636d62885f6cdab831cd97e5",
        "categoria": "adulto",
        "__v": 0
    }
}
```

API: POST: localhost:4000/api/registcomp/showallregristrocompetencia

Request:

```
{
    "competencia":"636d34bb032a4eff7077c70b"
}
```

Response:

```
[
    {
        "_id": "636d6c33170365890c213f0e",
        "competencia": "636d34bb032a4eff7077c70b",
        "jugador": "63420427b47797d72d2fe830",
        "categoria": "Open (15 y m s a os)",
        "__v": 0
    },
    {
        "_id": "636d6c6b170365890c213f32",
        "competencia": "636d34bb032a4eff7077c70b",
        "jugador": "634e04b5257cc9daadb685df",
        "categoria": "Infantil A1 (12 a os)",
        "__v": 0
    }
]
```

API:POST: localhost:4000/api/registcomp/GetOneregistro

Request:

```
{
    "_id":"636d6c33170365890c213f0e"
}
```

Response:

```
{
    "competition": {
        "_id": "636d6c33170365890c213f0e",
        "competencia": "636d34bb032a4eff7077c70b",
        "jugador": "63420427b47797d72d2fe830",
        "categoria": "Open (15 y m s a os)",
        "__v": 0
    }
}
```

API:DELETE: localhost:4000/api/registcomp/Deleteallcomp

Request:

```
{
    "competencia":"636d62885f6cdab831cd97e5"
}
```

Response:

```
{
    "competencia": "636d62885f6cdab831cd97e5",
    "competition": {
        "acknowledged": true,
```

```
        "deletedCount": 1
    }
}
```

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

A continuación se detallará una lista de conclusiones que se obtuvieron al finalizar el desarrollo del proyecto, estas conllevan desde aquellas de carácter tecnológico (usos de tecnologías) hasta otras de tipo más metodológico.

- La utilización de una herramienta cómo Oracle Cloud no se debe tomar a la ligera, lo ideal es tener al menos un poco de conocimiento básico previo sobre Linux, especialmente sobre la distribución CentOS, puede ser muy confuso y agobiante tratar de manejar desde la terminal todos este sistema operativo sin previamente tener algo de conocimiento del mismo ya que la posibilidad de ni siquiera entender que se está haciendo es grande.

- La elección de un *stack* de trabajo debe ser estudiada con antelación, ya sea que casi todos los miembros del equipo manejen las herramientas de dicho stack volviendo más fácil el inicio del desarrollo o que al menos el *stack* sea sencillo de aprender. Para cuestiones de este trabajo el equipo no tenía conocimientos previos en desarrollo web, por lo tanto se investigó los diferentes frameworks y *stacks* llegando así a la conclusión de que lo mejor era usar MERN, un*stack* de trabajo cuyas herramientas cuentan con demasiada información en internet (páginas, tutoriales, foros, ayudas, etc.) y aprenderlo es relativamente sencillo y rápido.

- En el desarrollo de un proyecto para un cliente verdadero con fecha de entrega se debe ser cuidadoso en el respeto a los tiempos de cada sprint y en el desarrollo del mismo ya que se está estableciendo un compromiso de entrega, es por esto que a la hora de planificar el sprint lo ideal es no exagerar con sus capacidades creyendo que se puede hacer más de lo que realmente el tiempo permite ni tampoco irse al otro extremo y entregar un mínimo que más bien parezca falta de compromiso e interés, se debe encontrar ese balance entre un buen *MVP* pero sin esclavizarse con el proyecto.

- Un trabajo tan grande que requiere tantos apartados no puede ser desarrollado por un equipo de trabajo que no se reparte desde el inicio sus roles y sus tareas. Es comprensible que no todos tienen las mismas capacidades pero sí todos deben buscar el cómo aportar o si no, por más que se intente el proyecto no saldrá a flote y no se cumplirán las reglas de cada sprint.

- Siempre que un apartado del proyecto conlleve mucha investigación lo ideal es que una vez que se llegue a la solución se documente inmediatamente todo lo investigado que sirvió para llegar a la solución, esto con el fin de que más adelante si es necesario aplicar otra vez el mismo

concepto no se tenga que investigar, sólo consultar notas.

- Desde el inicio del proyecto tenga claro que es un proyecto que será entregado a un cliente que posiblemente no sabe casi nada (o nada) de programación por lo tanto la entrega debe de ser un trabajo ”visible.ª los ojos del cliente, es decir, ojalá un proyecto funcional donde todo lo realizado en el desarrollo sea accesible por parte del cliente, por ejemplo en este caso el desarrollo fue una página web, pues entonces que a la hora de la entrega el cliente tenga una aplicación web, no un código, no un repositorio ni un *.zip*, si no una aplicación web visible en la que él puede entrar e interactuar.

6.2. Problemáticas y limitaciones

Problemáticas

A continuación se enlistan las problemáticas que el sistema presenta cómo errores, faltantes o comportamientos indeseables:

- En la utilización de filtros para saber que personas forman parte de una competencia esto se logra de una manera muy pesada y hace ver lento al sistema ya que toma la información de la competencia y luego va jugador por jugador verificando si está en la competencia o no, volviendo el método un tanto ineficiente.
- La utilización de çomprobantes de pago”(imagen de un comprobante) en el registro del delegdo no funciona ya que por alguna razón que el equipo no logra explicar al subirse la imagen al sistema y transformarlo a *base64* esta se guarda en la base de manera *corrupta* haciendo que a la hora de volver a tomar la imagen no sea visible o aparezca sólo media imagen o del todo no reconozca el archivo cómo imagen.
- El sistema no cuenta con *responsive design* para todos los dispositvos, es decir, si la aplicación web se visita desde un dispositivo cómo un celular o una tablet es muy posible que tenga una presentación pésima o que simplemente ni se pueda usar porque todo se verá desordenado.
- La base de datos a usar está ligada a una cuenta de Mongo Atlas y no en una base de datos interna dentro de la máquina virtual haciendo que el sistema no solo dependa de la máquina virtual si no de un componente externo. Esto se debe a que por más que se investigó no se llegó a la solución de cómo instalar Mongo DB versión 6 en Linux CentOS.
- El sistema permite que una competencia se cree con una fecha anterior a la actual lo cual carece de sentido.

Limitaciones

En este apartado se detallan las limitaciones que tiene el sistema, es decir, aquellas cosas que el sistema no puede hacer o que se querían hacer pero no se pudieron incluir. Además también se muestra aquí mismo las plataformas para las que está diseñado y se comprobó que en ellas funciona cómo se esperaba.

- El sistema de envío de correo desde la página para avisar a un delegado el porqué fue rechazado.
- Desarrollo de la interfaz con *responsive design* para que funcione en todos los dispositivos.
- Animaciones para que la interfaz se sienta más amigable.

- **Plataformas en que funciona**
  - Computadoras con Windows y Linux (no recomendable de usar en dispositivos móviles o tablets).
  - Navegadores web:
    - Google Chrome
    - Mozilla Firefox
    - Opera
    - Microsoft Edge
  - Requerimientos mínimos:
    - Conexión a internet
    - Navegador web instalado (vease los anteriores mencionados)
    - Computadora con capacidad de abrir un navegador de los anteriormente mencionados.
      - ◇ Un procesador Intel Pentium 4 o superior compatible con SSE3.
      - ◇ Memoria RAM 512Mb o superior.

6.3. Trabajo futuro

A continuación se detallan algunos trabajos que se considera que se pueden mejorar en el futuro de este proyecto.

- Creación de un sistema de correo con bandeja de entrada y salida para una comunicación más sencilla entre delegado-administrador y administrador-delegado.
- Implementación de *responsive design* para que sea accesible desde cualquier dispositivo.
- Animaciones para que la interfaz se sienta más amigable.
- Implementar todo el proyecto en un sistema de *Oracle Cloud* de paga para poder usar más recursos de la máquina virtual.
- Para los comprobantes arreglar el tema de las imágenes y además que acepte otros tipos de archivos cómo PDF.
- Mejorar la interfaz gráfica a una más estética y minimalista.
- Incluir un método para conectarse con el WhatsApp de la FEDEPAT para así tener diferentes medios de comunicación.
- Volver La creación de tipos de torneo más flexible, es decir, que el administrador pueda escribir el tipo de torneo y agregarlo a las opciones de tipos.
- Agregar la opción a la creación de competencias que además de la provincia de la competencia se seleccione el cantón y el distrito.
- Crear filtros para los elementos agregados en el punto anterior.
- Que las competencias tengan fecha de inicio, fecha final y número máximo de cupos.

Referencias bibliográficas

CodigoFacilito. (2022). Que es mongoose. Descargado 18 de agosto 2022, de https://codigofacilito.com/articulos/que-es-mongoose

ExpressJS. (2022). Express - infraestructura de aplicaciones web node.js. Descargado 18 de agosto 2022, de http://expressjs.com/es/

NodeJS. (2022). Acerca — node.js. Descargado 15 de agosto 2022, de https://nodejs.org/es/ about/

ReactJS. (2022). React, una biblioteca de javascript para construir interfaces de usuario. Descargado 15 de agosto 2022, de https://es.reactjs.org/