|  |
| --- |
|  |
| Proyecto final Desarrollo de Aplicaciones Web |
| Documentación |
|  |
| **DANIEL TORRENS GONZÁLEZ** |
| **21/03/2021** |

|  |
| --- |
| Documentación sobre el desarrollo del proyecto final del ciclo de grado superior Desarrollo de Aplicaciones Web. Consistirá en una aplicación que se encargará de gestionar torneos de ajedrez. Utilizaremos los contenidos dados acerca del front-end y el back-end, conexión a una base de datos y, por último, habilitar la aplicación en la nube para hacerla disponible para el resto de posibles usuarios. |

ÍNDICE

[1. Introducción 2](#_Toc67262527)

[1.1 Presentación y objetivos 2](#_Toc67262528)

[1.2 Contexto 2](#_Toc67262529)

[1.3 Planteamiento del problema 3](#_Toc67262530)

[1.4 Estructura del documento 3](#_Toc67262531)

[2. Análisis de requerimientos (con una valoración en horas) 4](#_Toc67262532)

[3. Diseño del sistema 5](#_Toc67262533)

[4. Implementación 6](#_Toc67262534)

[5. Validación y pruebas 7](#_Toc67262535)

[6. Mantenimiento 8](#_Toc67262536)

# Introducción

# Presentación y objetivos

Este documento describe el trabajo realizado en el proyecto final del ciclo de grado superior Desarrollo de Aplicaciones Web. El proyecto consiste en una aplicación que se encargue de gestionar torneos de ajedrez. Se centrará en los torneos que puedan desarrollarse en la isla principalmente y se prestará atención a los jugadores y a un registro de las partidas disputadas.

El objetivo de dicha aplicación es dar una información completa sobre el torneo en cuestión, los jugadores que lo disputaron, sus resultados, los árbitros que arbitraron cada partida y los movimientos de cada partida para futuros análisis. La web es accesible desde cualquier navegador por Internet y, en cuanto a los usuarios, la aplicación distingue tres tipos. Por un lado, están los usuarios no registrados, que únicamente pueden acceder a información general sobre el torneo. Luego nos encontramos los usuarios registrados que también podrán mirar las partidas y los movimientos realizados en la misma por ambos jugadores. Por último, podemos encontrar los administradores, los cuales no solo tienen todas las funcionalidades de los dos tipos anteriores sino que además son los únicos que pueden crear nuevos torneos, añadir o quitar árbitros y jugadores.

# Contexto

El proyecto ha sido realizado para el Club Deportivo CIAL (Centro Insular de Ajedrez de Lanzarote) situado en la capital de la isla, Arrecife, en la provincia de Las Palmas, España. Para el desarrollo de la aplicación se realizaron diversas reuniones con la dirección del club y estudié cómo se estaban realizando los torneos para después poder informatizarlos adecuadamente. La mayoría de las decisiones de estética y funcionales de la aplicación fueron tomadas junto con los jugadores federados, que además me proporcionaron toda la documentación del club. Una vez realizadas las reuniones con la directiva y los jugadores, se distribuyó el trabajo en 3 partes: el diseño de la aplicación web, la intranet de los usuarios y la del administrador.

# Planteamiento del problema

El problema que se nos ha planteado es la construcción del sitio web para distintos torneos. La funcionalidad de la aplicación, a grandes rasgos, debía ser:

* Mostrar información general de cada torneo
* Permitir a los usuarios mirar la información de cada torneo y cada jugador
* Permitir a los usuarios registrados visualizar las partidas
* Permitir a los administradores la creación, el borrado y la actualización de los torneos así como de los jugadores y árbitros que las disputen.

# Estructura del documento

El presente documento está dividido en una serie de capítulos que corresponden, básicamente, a las distintas etapas que conforman el proceso de desarrollo del proyecto. Estas etapas han sido:

- Análisis de requerimientos: Se redactó de una manera global una primera visión del proyecto donde señalamos los requisitos que debía cumplir. La finalidad de esta etapa es plasmar el acuerdo entre el desarrollador y el cliente acerca de las funcionalidades del proyecto. En nuestro caso el visto bueno nos lo dio la dirección del CIAL. Se realizó el modelado conceptual de la futura solución mediante el uso de diagramas (diagrama de clases, diagramas de casos de uso, diagrama de entidad relación). Los modelos ayudan a visualizar como es el sistema, proporcionando plantillas que sirven de guía en la construcción de la aplicación. En esta etapa se especifica qué debe hacer la aplicación pero no cómo debe hacerlo.

- Diseño del sistema: Se utilizaron los elementos y modelos obtenidos durante el análisis para transformarlos en mecanismos que puedan ser utilizados en un entorno web con las características y condiciones que establecen este tipo de entornos. Se diseñaron todos los niveles de los que consta la aplicación (nivel de presentación, nivel lógico y nivel de persistencia). Tanto la etapa del análisis como la del diseño están desprovistas de código. Un buen análisis y un buen diseño son la mejor forma de llegar a producir software de calidad.

- Implementación: Se utilizaron los elementos obtenidos en el diseño para permitir la elaboración del producto o prototipo funcional, es decir, que puede ser puesto en marcha y sometido a pruebas. Para ello se consideraron las diversas tecnologías que han intervenido en la elaboración de dicho producto. Todo lo desarrollado en las etapas del análisis y del diseño, se tradujo a código.

- Valoración y pruebas: Esta fase se centró en la comprobación del correcto funcionamiento del producto desarrollado mediante una serie de pruebas.

- Mantenimiento: En esta última fase se trata de tener una idea sobre posibles copias de seguridad, habilitar el servidor para que puedan acceder a la aplicación y posibles políticas de actualizaciones en función de lo que el cliente pueda sugerir en el futuro.

Tras estas tareas pudimos dar por concluido el proyecto, por lo que en último lugar mostramos las conclusiones obtenidas y listamos la bibliografía utilizada durante la realización del mismo.

# Análisis de requerimientos (con una valoración en horas)

Se desea realizar la creación e implementación de una herramienta de software para almacenar los participantes de diferentes torneos y/o campeonatos de ajedrez, los árbitros correspondientes a cada evento, el equipo al que pertenecen los jugadores y el país de donde provienen tanto los jugadores como los árbitros.

En segundo lugar, se quiere llevar un seguimiento de cada una de las partidas que se disputan en cada torneo y/o campeonato, así como un registro de cada evento y, por supuesto, el registro de cada movimiento (las jugadas) de cada partida.

Por último, se desea llevar los movimientos de las partidas a un tablero y poder reproducirla, escogiendo el campeonato y luego todas las partidas jugadas en él, o buscando a un jugador en concreto y pudiendo elegir cualquier partida que haya jugado independientemente del evento en el que se disputó.

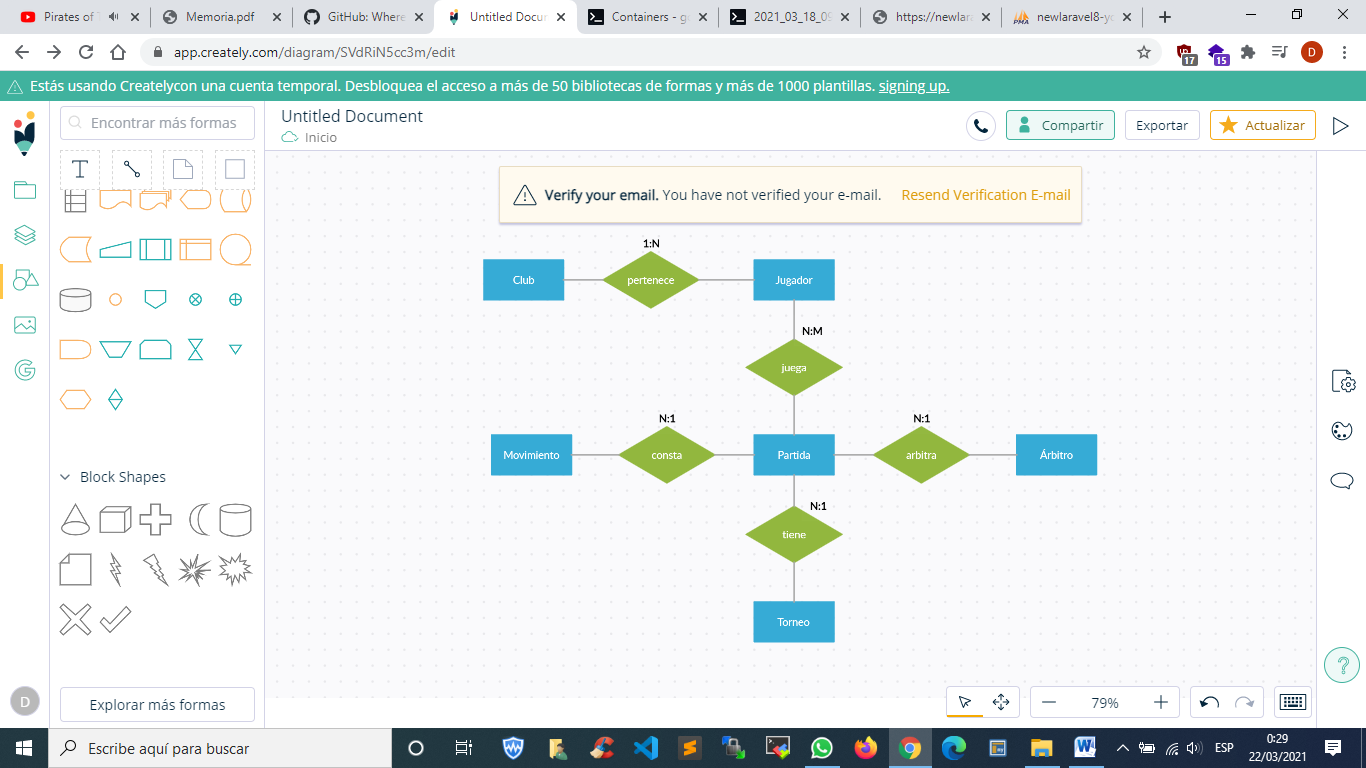
La aplicación podrá distinguir entre usuarios y administradores. Los primeros tendrán la capacidad de ver las partidas y reproducirlas, buscar jugadores y árbitros así como torneos, y mirar que clubes y países envían jugadores. Sin embargo, los administradores además de todo lo anterior, también pueden crear un nuevo evento y añadir jugadores y árbitros. En caso de añadir un club o un país participante, los administradores serán los encargados de hacerlo.

Finalmente, una persona que acceda a la aplicación deberá registrarse y según sea usuario o administrador será capaz de hacer unas u otras funciones especificadas en párrafos anteriores.

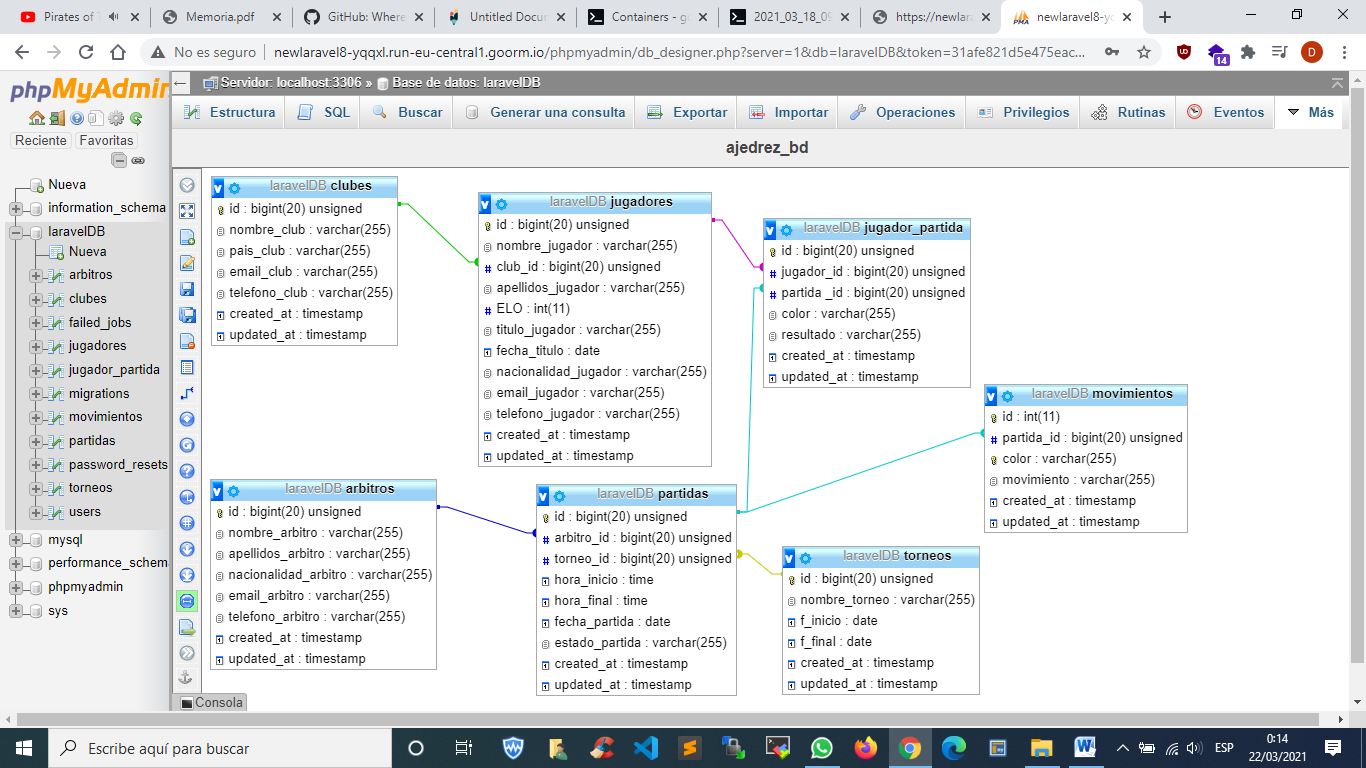
Atendiendo ahora al tiempo necesitado para poder llevar a cabo este proyecto, calculando aproximadamente un tiempo de 14 días a 8 horas trabajadas por día, y teniendo en cuenta que un desarrollador freelance junior puede cobrar la hora a 20€, tenemos:

# Diseño del sistema

En cuanto al diseño del sistema, empezaremos centrándonos en el modelo entidad-relación. La base de datos estará centrada en la entidad Partida, de la cual girará el resto de entidades que necesitaremos. Como podemos ver, de la entidad Partida colgarán directamente cada movimiento de la entidad Movimiento, cada jugador de la entidad Jugador, cada árbitro de la entidad Árbitro y cada torneo de la entidad Torneo, teniendo una relación de “muchos a muchos” entre la entidad Jugador y la entidad Partida. Por último, a cada jugador le corresponde un club de la entidad Club.



Las tablas derivadas de la imagen anterior quedarían de la siguiente manera relacionadas, cada una con sus correspondientes campos:



# Implementación

La implementación del código se llevará a cabo en el framework Laravel, versión 8, ya que utilizaremos php como lenguaje de programación del lado del servidor. Además, en las plantillas que nos proporciona el propio Laravel podemos emplear HTML, CSS y JavaScript para el lado del cliente.

Utilizaremos una base de datos usando la herramienta phpmyadmin y MySQL como gestor de base de datos. Además, desde el propio Laravel modificaremos las tablas de la base de datos y los datos a insertar en cada campo si fuese necesario.

A todo lo anterior se suma todas las librerías que nos sean necesarias para añadirlas al proyecto y generar, entre otros, pdf, códigos QR, formato de fechas, validaciones de formularios, registros, gráficas, etc.

Por último, la aplicación se desarrollará utilizando el framework Laravel que, a su vez, lo instalaremos y configuraremos dentro de un IDE en la nube llamado goormide.

Cabe mencionar que dentro del goormide existen distintos paquetes de precios para poder desarrollar mejor los proyectos que se nos encarguen. Así, la versión Premium que nos otorgaría más y mejores funcionalidades cuesta 14.40 $ al mes, lo que supone un total de 172.80 $ al año, lo equivalente a 145.36 €.

# Validación y pruebas

En el caso que nos ocupa, el proyecto se realizará en el framework llamado Laravel, utilizado para el desarrollo de aplicaciones web con php como lenguaje de programación de parte del servidor.

Comentaremos también que empleamos lo que se conoce como Test-Driven Development (TDD) que no es más que una práctica de ingeniería para de software cuyo propósito es el de un desarrollo guiado por pruebas para lograr un código limpio que funcione. TDD involucra otras dos prácticas las cuales son: escribir las pruebas primero (Test First Development) y la Refactorización.

Es posible la realización de pruebas unitarias (unit test) en Laravel. Estas pruebas son una herramienta para detectar de una manera rápida y en fase de desarrollo si nuestro código está generando algún bug y cumple la lógica esperada.

Las pruebas unitarias son tests que atacan a una parte específica del sistema y son muy rápidas. Por eso podemos tener una gran cantidad de pruebas de esta clase. Por último, para aplicar estos tests debemos tener en cuenta que: especificar los parámetros de entrada, cuáles son los valores que retornan las funciones/métodos y atender con especial detalle a los bucles y condicionales.

En el Laravel podemos hacer esto con el comando:

*php artisan make:test rutaDeLaClase/nombreDeLaClaseTest*

Y la clase debe acabar con la palabra Test. Ahora para poder ejecutarlos empleamos los siguientes comandos (ejecutará todos los tests de tu proyecto, ejecutará solo un test y ejecutará todos los tests de una clase, respectivamente):

*vendor/bin/phpunit*

*vendor/bin/phpunit --filter methodName path/to/file.php*

*vendor/bin/phpunit --filter className*

También podemos utilizar las distintas librerías que Laravel ofrece como por ejemplo Laravel Dusk. Este componente nos permitirá probar la aplicación desde la perspectiva de un usuario, incluso aunque usemos JavaScript en el navegador. Podemos instalarlo, crear el directorio de tests y ejecutar los tests respectivamente con los siguientes comandos:

*composer require laravel/dusk*

*php artisan dusk:install*

*php artisan Dusk*

# Mantenimiento

El mantenimiento de la aplicación será realizado por el propio desarrollador, atendiendo a futuras versiones dependiendo de los añadidos que quiera el cliente. El desarrollador mantendrá operativo todo lo necesario para que la aplicación pueda ser accesible desde fuera del entorno goormide, esto es, instalar y configurar un servidor y abrir los puertos necesarios para su comunicación con el exterior.

En cuanto a las mejoras posibles que puedan suponer un cambio en la aplicación, se propone poder jugar partidas entre dos usuarios y analizar dichas partidas. Todo esto sin que se pierda el acceso a la aplicación y su funcionamiento básico detallado en el punto primero de este documento.

También debemos señalar que Laravel cuenta con una librería llamada Laravel BackUp que nos permite comprimir todos nuestros ficheros en un fichero.zip y hace un volcado de todos los datos que conforman nuestra base de datos en un fichero.sql. Todo esto podemos luego moverlo a nuestro ordenador local o a otras máquinas para tenerlas como copias de seguridad.