

Universidad de San Carlos De Guatemala
Facultad de ingeniería
Escuela de Ciencias y Sistemas
Manejo e Implementación de Archivos



Proyecto 2

Manual de Usuario-Tecnico

Edson Saul Avila Ortiz
201902302

Quinela App

Quinela app es una aplicación dividida en dos partes con el objetivo de la administración, ejecución y uso de un sistema de apuestas o quinielas deportivas. Consta de 3 módulos fundamentales los cuales son:

- 1) Home
- 2) Administrador
- 3) Usuario

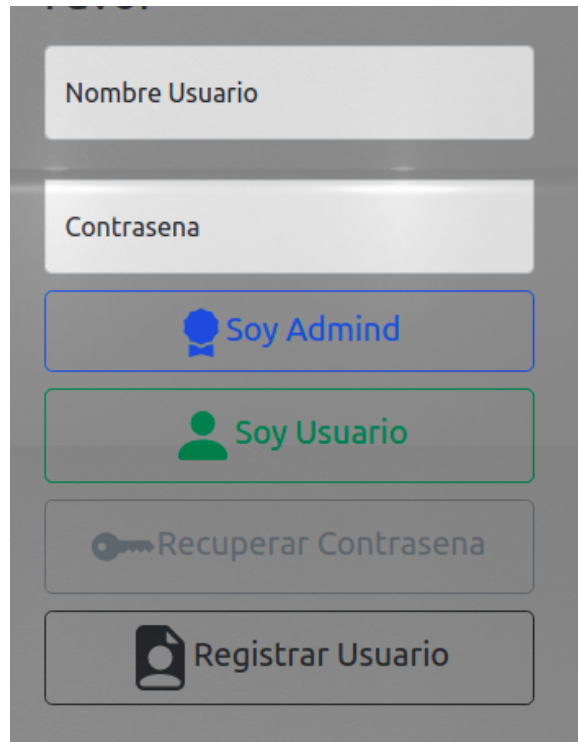
Home

Una vez entramos a la aplicación este es el apartado que se nos mostrará dentro del cual tenemos 4 opciones diferentes cada una con diferente funcionalidad y objetivo específico.



- recuperar contraseña: este apartado nos permite recuperar la contraseña de cualquier usuario independientemente si éste es administrador, los datos necesarios para hacer este proceso únicamente son el nombre de usuario y la contraseña que desea ingresar dependiendo de la naturalidad o el resultado de la transferencia se mostrará un mensaje del usuario.
- Login Usuario: este apartado es el login para todos los usuarios que no sean administradores dependiendo de los datos que hayan sido enviados y la transferencia que se haya realizado se mostrará un mensaje de aceptado o no, si el mensaje fue aceptado es decir que el usuario ingresar los datos correctos se le redirigirá a una nueva ventana donde se encontrarán todas las opciones para este usuario.
- Login Administrador: el login del administrador se hace en un módulo aparte debido a la naturalidad del mismo ni los diferentes componentes extras que se le tiene que demostrar al administrador para llevar un mejor control sobre el funcionamiento global del programa al igual que el login de usuario mostrar un mensaje del resultado de la transferencia es redirigirá al apartado específico.

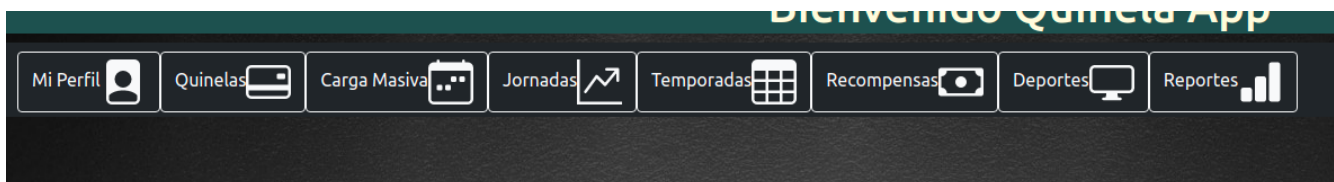
- registro usuario: en este apartado se le permite a cualquier usuario con acceso a la aplicación puede registrarse dentro de la misma para validar el registro exitoso de un usuario se necesitan ciertas restricciones las cuales vienen siendo: una contraseña válida, correo válido, edad aceptada, nombre de usuario único. si los datos que ingreso el usuario no cumplen con estas normativas el trigger de la aplicación regresará un error el cual se demostrará el usuario y tendrá que rellenar los datos necesarios para poder registrarse.



Formulario de registro de usuario con los siguientes campos y botones:

- Campo de texto: Nombre Usuario
- Campo de texto: Contraseña
- Botón: Soy Admin (con icono de corona)
- Botón: Soy Usuario (con icono de persona)
- Botón: Recuperar Contraseña (con icono de llave)
- Botón: Registrar Usuario (con icono de documento)

Administrador



Perfil

En este apartado se muestran los datos del administrador así como su nombre de usuario, su nombre y apellido además de permitirle cambiar la contraseña.

carga masiva

en este apartado se realizan las operaciones de carga masiva del archivo mostrar un mensaje de retorno dependiendo el resultado de la transferencia, mientras esa transferencia no haya sido completada se mostrará un mensaje para que el usuario Espere.

Jornadas

En esta sección se realizan todas las tareas relacionadas con jornada las cuales vienen siendo, mostrar información de la jornada actual, editar la jornada actual así como su fecha y hora de finalización, crear eventos deportivos dentro de la jornada y asignar resultados de la jornada.

Temporada

en esta sección se muestra la tabla de posiciones de la temporada actual ordenada por puntaje descendente mostrando resultados de todos los usuarios

recompensas

esta pantalla muestra información sobre las recompensas monetarias que han sido otorgadas a los usuarios a lo largo de las temporadas. al entrar a esta posición se muestra una tabla con todos los usuarios registrados en el sistema junto con el total de dinero que han acumulado ordenados por el total de su recompensa descendiente.

Deportes

en este apartado se muestra un listado de todos los deportes que han sido registrados en todas las temporadas además de su información básica como el nombre, color e imagen que los identifican.

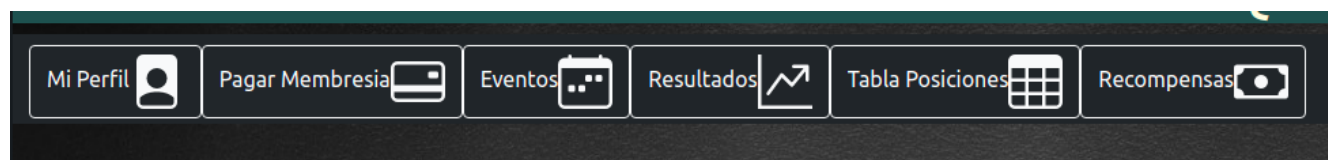
Reportes

en este apartado se muestran los diferentes reportes generados en la aplicación las cuales vienen siendo 5.

Chat

este apartado permite iniciar arrendar un chat entre el administrador y algún usuario.

Usuario



Perfil

en este apartado se muestran los datos del usuario, se muestra su nombre apellido, su nombre de usuario y además le permite cambiar la contraseña del usuario en este mismo apartado sin necesidad de salir de la aplicación.

Pagar membresía

en este apartado se le permite al usuario poder para su membresía para la siguiente temporada o bien cambiar la membresía que éste posee, al usuario se le permite cambiar entre 3 diferentes membresías las cuáles son: oro, plata y bronce.

Eventos

en este apartado se le permite al usuario poder ingresar predicciones a la base de datos dependiendo de los eventos que se encuentren activos actualmente

Resultados

en este apartado se muestran los resultados obtenidos por el usuario a lo largo de las temporadas señalando las predicciones en las que acertado y el total de recompensas que se le ha dado.

Manual técnico

El objetivo de este documento es documentar de forma técnica el funcionamiento de la aplicación para facilitar su posterior modificación, actualización o cambios que se le quieran hacer a la misma; Asimismo de especificar las tecnologías que se utilizaron para poder levantar los servicios necesarios para funcionamiento de la aplicación.

Objetivos

- Implementar un modelo de base de datos eficaz y escalable.
- Crear una plataforma web con una conexión a una base de datos relacional.
- Construir una base de datos a través del uso de archivos de scripts SQL.
- Construir sistemas funcionales a través de React y Golang.
- Diseñar un sistema funcional y atractivo para negocios reales.

Requerimientos:

- Instalación de go en la versión indicada
- instalación un manejador de paquetes como NPM
- instalar todas las dependencias de go y react utilizadas en el proyecto
- Instalación del cliente de Oracle si se desea acceder a la aplicación
- instalación de Oracle development y sus dependencias

Tecnología Utilizada:

- Oracle

Para almacenamiento de todos los datos necesarios dentro de la aplicación se optó para usar una base de datos relacional en oráculo en la cual almacena funciones, procedimientos y triggers para validar la integridad de las transferencias entre la aplicación y la base de datos.

- Go: se utiliza una red levanta en el lenguaje go versión 1.14

Go es expresivo, conciso, limpio y eficiente. Sus mecanismos de concurrencia facilitan la escritura de programas que aprovechan al máximo las máquinas multinúcleo y en red, mientras que su novedoso sistema de tipos permite la construcción de programas flexible y modular. Go compila rápidamente en código de máquina, pero tiene la conveniencia de la recolección de basura y el poder de la reflexión en tiempo de ejecución. Es un lenguaje compilado, rápido, tipado estáticamente que se siente como un lenguaje interpretado dinámicamente tipado.

- React: se utilizó la versión 16.8 para levantar un frontend adecuado para la aplicación

React te ayuda a crear interfaces de usuario interactivas de forma sencilla. Diseña vistas simples para cada estado en tu aplicación, y React se encargará de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien.

Librerías Externas:

- Godor

godror es un paquete que es una base de datos / sql / driver.Driver para conectarse a Oracle DB, utilizando el excelente contenedor OCI de ODPI-C.

Installation

Run:

```
go get github.com/godror/godror
```

Then install Oracle Client libraries and you're ready to go!

See [Godror Installation](#) for more information.

Connection

To connect to Oracle Database use `sql.Open("godror", dataSourceName)`, where `dataSourceName` is a [logfmt](#)-encoded parameter list. Specify at least "user", "password" and "connectString". For example:

```
db, err := sql.Open("godror", `user="scott" password="tiger" connectString="dbhost:1521/orclpdb1"`)
```

The `connectString` can be *ANYTHING* that SQL*Plus or Oracle Call Interface (OCI) accepts: a service name, an [Easy Connect string](#) like `host:port/service_name`, or a connect descriptor like `(DESCRIPTION=...)`.

For more connection options, see [Godor Connection Handling](#).

- Mux

El paquete gorilla / mux implementa un enrutador de solicitudes y un despachador para hacer coincidir las solicitudes entrantes con su respectivo controlador.

Install

With a [correctly configured](#) Go toolchain:

```
go get -u github.com/gorilla/mux
```

- Handlers

Los controladores de paquetes son una colección de controladores (también conocidos como "middleware HTTP") para usar con el paquete net / http de Go (o cualquier marco que admita `http.Handler`)

Example

A simple example using `handlers.LoggingHandler` and `handlers.CompressHandler`:

```
import (
    "net/http"
    "github.com/gorilla/handlers"
)

func main() {
    r := http.NewServeMux()

    // Only log requests to our admin dashboard to stdout
    r.Handle("/admin", handlers.LoggingHandler(os.Stdout, http.HandlerFunc>ShowAdminDashboard)))
    r.HandleFunc("/", ShowIndex)

    // Wrap our server with our gzip handler to gzip compress all responses.
    http.ListenAndServe(":8000", handlers.CompressHandler(r))
}
```

License

BSD licensed. See the included LICENSE file for details.

- react-calendar

React-Calendar está en constante desarrollo. Esta documentación está escrita para la rama React-Calendar 3.x. Si desea ver la documentación de otras versiones de React-Calendar, use el menú desplegable en la parte superior de la página de GitHub para cambiar a una etiqueta apropiada.

Installation

Add React-Calendar to your project by executing `npm install react-calendar` or `yarn add react-calendar`.

Usage

Here's an example of basic usage:

```
import React, { useState } from 'react';
import Calendar from 'react-calendar';

function MyApp() {
  const [value, onChange] = useState(new Date());

  return (
    <div>
      <Calendar
        onChange={onChange}
        value={value}
      />
    </div>
  );
}
```