



Proyecto 2

11/05/2021

Wilfred Stewart Perez Solorzano

Ingenieria en Ciencias y Sistemas

Manejo e implementacion de archivos seccion A-

Primer Semestre 2021

Visión general

La empresa TodoDeporte GT , quienes eran conocidos en la escena deportiva por la organización de quinielas en

eventos deportivos importantes se han dado cuenta de esto y desean entrar al negocio de quinielas en la web. Por

tal razón, se le contrata a usted, con sus conocimientos en bases de datos y aplicaciones web para que diseñe un

portal web que implemente el proceso entero de una quiniela.

Objetivos

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

Especificaciones

Lenguajes y herramientas a utilizar por el proyecto

- Para el backend se utilizará el lenguaje de programación Go (Golang).
- Para el frontend se utilizará el framework web React.
- HTML5 y CSS para que el sitio sea funcional, atractivo y refleja la imagen de cada entidad.
- Oracle en cualquier versión mayor o igual a 12c en su versión Express para la base de datos. El método de
- La instalación queda a discreción del estudiante.

Procesos

Procesos de entrada

Programación de escritorio

1. Ingresar al programa de escritorio (acceso).
2. Ingresar datos para el registro de productos (ingreso, devoluciones, perdidas)
3. Ingresar datos para registros de usuarios (cliente, empleado). Ingresar datos para registros de ventas.

Consultas:

- Consulta nuevo usuario (usuario, administrador)
- Cambio de información (usuario, administrador)
- Carga de archivo masivo (administrador)
- Obtención general de datos (administrador)

Procesos de salida

Programa de escritorio

- Consulta de credenciales (Usuario, Administrado).
- Consulta de nuevos usuarios (Usuario).
- Consulta de formulario de informacion (usuario, administrado).

Base de Datos Oracle

Utilización de Docker para la utilización de un contenedor de oracle y tenerlo localmente en el sistema operativo Linux.

Requisitos del sistema

- Requerimientos de hardware
 - Equipo, teclado, mouse, monitor
 - Memoria RAM 2 GB
 - Tarjeta de red LAN y/o Wireless
 - Procesador 1.4 GHz.

- Requerimientos de software
 - Sistema operativo Ubuntu (recomendado 20.04)
 - Conexión internet local



Herramientas utilizadas

Go

En septiembre de 2007, tres programadores de la plantilla de Google, Robert Griesemer, Rob Pike y Ken Thompson, presentaron sus ideas para un lenguaje de programación optimizado y más sencillo, sentando así las bases de Go. Lo que comenzó siendo un **proyecto modesto** se convirtió pronto en un ambicioso propósito cuyo desarrollo se vio **impulsado por Google**, que dispuso todos los recursos necesarios para ello. Desde que se publicara oficialmente como proyecto **open source** con licencia BSD a finales de 2011, a Go no le ha sido difícil rodearse de una amplia comunidad que hasta el día de hoy no ha dejado de colaborar activamente en su continuo desarrollo y en su optimización. La primera versión estable (1.0) se publicó el 28 de marzo de 2012 y desde la versión 1.1 (un año después) Google publica actualizaciones cada medio año.

Servidor de base de datos (Oracle)

Oracle Support puede proporcionar exactamente la cobertura de soporte que necesita para el éxito de su negocio. Ya sea que su entorno de Oracle sea local, en la nube o en una nube híbrida, proporcionamos soporte y orientación de un extremo a otro, cuándo, cómo y dónde lo necesite.

Docker

Docker ayuda a los desarrolladores a dar vida a sus ideas al conquistar la complejidad del desarrollo de aplicaciones. Simplificamos y aceleramos los flujos de trabajo de desarrollo con un canal de desarrollo integrado y mediante la consolidación de componentes de aplicaciones. Utilizados activamente por millones de desarrolladores en todo el mundo, Docker Desktop y Docker Hub brindan una simplicidad, agilidad y opciones incomparables.

Inicialización

Requisitos generales: conexión a internet

Ejecutar mediante una consola de linux, la aplicación de escritorio con el comando “npm start” dentro de la carpeta “myapp” del proyecto en cuestión

```
willop@willop-H81MLV3:~/Documentos/USAC/Archivos/Proyecto2/Proyecto2/src/myapp$ npm start
```

Seguido abrir otra terminal en la raíz del proyecto y ejecutar el siguiente comando, “go run main.go FuncionImagen.go”

```
willop@willop-H81MLV3:~/Documentos/USAC/Archivos/Proyecto2/Proyecto2/src$ go run main.go FuncionConvert.go
```

Con lo cual se está inicializando el sitio web y el backend de la aplicación

Ejecución de front end.js (React)

```
src/Views/CrearUsuario.js
  Line 12:8:  'Cookies' is defined but never used          no-unused-vars
  Line 91:21: img elements must have an alt prop, either with meaningful text, or an empty string for decorative images  jsx-ally/alt-text

src/Views/Login.js
  Line 10:8:  'imgg' is defined but never used          no-unused-vars
  Line 67:21: img elements must have an alt prop, either with meaningful text, or an empty string for decorative images  jsx-ally/alt-text

src/Views/Usuario.js
  Line 2:10:  'Input' is defined but never used          no-unused-vars
  Line 2:17:  'ButtonToggle' is defined but never used   no-unused-vars
  Line 8:8:   'imgg' is defined but never used          no-unused-vars
  Line 79:7:  React Hook useEffect has a missing dependency: 'imgg'. Either include it or remove the dependency array  react-hooks/exhaustive-deps

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.
```

Ejecucion del backen.go

```
willop@willop-H81MLV3:~/Documentos/USAC/Archivos/Proyecto2/Proyecto2/src$ go run main.go FuncionConvert.go
Servidor de GO execute
Port:4000
...
```

Para la base de datos

Visualización de contenedores activos

```
willop@willop-H81MLV3:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Inicialización de contenedor de la base de datos

```
willop@willop-H81MLV3:~$ docker start musing_black
musing_black
```

Ejecucion del docker

```
willop@willop-H81MLV3:~$ docker exec -it musing_black bash
```

Instalacion de la base de datos en el sistema

```
willop@willop-H81MLV3:~$ docker exec -it musing_black bash
```

Ejecucion de instalación de la base de datos de oracle

```
Actividades Terminal 11 de may 23:25
willop@willop-H81MLV3: ~
Setting ENV
Starting default listener
Parsing command line arguments:
  Parameter "silent" = true
  Parameter "responsefile" = /u01/app/oracle/product/18.0.0/dbhome_1/netca.rsp
Done parsing command line arguments.
Oracle Net Services configuration:
Configuring Listener:LISTENER_ORCL18
Listener configuration complete.
Oracle Net Listener Startup:
Running Listener Control:
/u01/app/oracle/product/18.0.0/dbhome_1/bin/lsnrctl start LISTENER_ORCL18
Listener control complete.
Listener started successfully.
Profile configuration complete.
Oracle Net Services configuration successful. The exit code is 0
Configuring the TNS
Testing Database
SQL*Plus: Release 18.0.0.0.0 - Production on Wed May 12 03:22:59 2021
Version 18.3.0.0.0
Copyright (c) 1982, 2018, Oracle. All rights reserved.
Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area 1610612704 bytes
Fixed Size 8896480 bytes
Variable Size 536870912 bytes
Database Buffers 1056964608 bytes
Redo Buffers 7880704 bytes
Database mounted.
Database opened.
SQL>
System altered.

SQL> alter pluggable database PDB18C open
*
ERROR at line 1:
ORA-65019: pluggable database PDB18C already open

SQL>
NAME OPEN_MODE
-----
ORCL18 READ WRITE

SQL>
CON_ID CON_NAME OPEN_MODE RESTRICTED
-----
2 PDB$SEED READ ONLY NO
3 PDB18C READ WRITE NO
SQL>
SQL*Plus: Disconnected from Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production
```


Entidad de relación

