

Aplicación web para la administración de empleados

Xavier Garzón López
xavgar9@gmail.com
github.com/xavgar9

Índice General

Descripción del Problema.....	2
Planteamiento del problema.....	3
Objetivos.....	3
Objetivo General.....	3
Objetivos Específicos.....	3
Obtención y Análisis de Requisitos.....	4
Vista General del Software.....	4
Diagrama de casos de uso.....	4
Requisitos.....	4
Detalles de los Requisitos.....	5
Nota 1.....	5
Nota 2.....	6
Diseño y Tecnologías de Software.....	7
Arquitectura.....	7
Tecnologías de Software.....	7
Pruebas de Software.....	8
Pruebas Unitarias.....	8

Capítulo 1

Descripción del Problema

1.1 Planteamiento del problema

Contar con herramientas tecnológicas que apoyen los procesos dentro de las empresas es fundamental para garantizar su correcto funcionamiento. En este momento, la empresa realiza el registro y salida de empleados, junto con la administración de los mismos, con hojas de cálculo que han dificultado y retrasado a los empleados.

Es importante usar un software que esté al nivel de las exigencias de la empresa y no entorpezca sus labores.

1.2 Objetivos

1.2.1 Objetivo General

Diseñar y construir una aplicación web que permita realizar administrar empleados.

1.2.2 Objetivos Específicos

- Identificar y analizar los aspectos más importantes del software.
- Establecer y especificar los requisitos del software.
- Diseñar la arquitectura del software.
- Desarrollar el software que cumpla con los requisitos establecidos para el mismo.
- Evaluar el software mediante pruebas unitarias.

Capítulo 2

Obtención y Análisis de Requisitos

2.1 Vista General del Software

Este aplicativo web es un software pequeño que permite administrar usuarios de manera básica. Las cuatro operaciones básicas que se pueden realizar son las siguientes:

- Crear usuarios
- Visualizar usuarios
- Editar usuarios
- Eliminar usuarios

2.2 Diagrama de casos de uso

En la figura 2.1 se puede observar cómo un usuario puede realizar distintas acciones dentro del sistema para poder administrar los empleados.

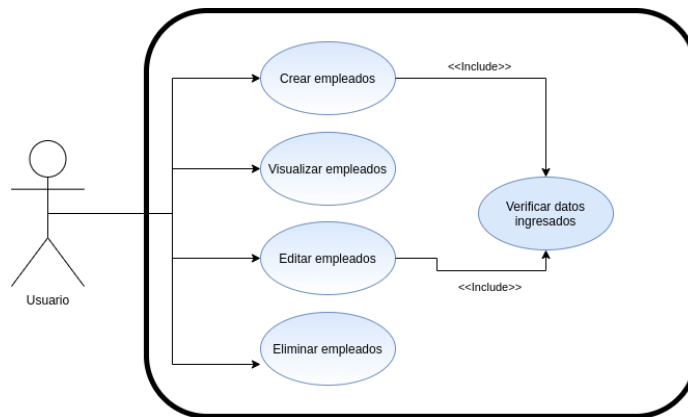


Figura 2.1: diagrama de casos de uso del software.

2.3 Requisitos

Para obtener los requisitos del sistema, primero se realizó un análisis de las exigencias de la empresa. Basado en ello, se obtuvieron los siguientes requisitos.

ID	Nombre	Descripción	Dependencias	Notas
EM-01	Registrar empleados	El sistema debe permitir a los usuarios registrar empleados nuevos	-	Ver nota 1
EM-02	Consultar empleados	El sistema debe permitir a los usuarios consultar empleados	EM-01	-
EM-03	Editar empleados	El sistema debe permitir a los usuarios editar empleados	EM-01	Ver nota 2
EM-04	Eliminar empleados	El sistema debe permitir a los usuarios eliminar empleados después de una confirmación de la acción.	EM-01	-

Figura 2.2: requisitos del software.

2.4 Detalles de los Requisitos

2.4.1 Nota 1

Al momento de registrar empleados nuevos se deben cumplir las siguientes condiciones con sus atributos.

Nota 1				
Atributo	Restricciones	Longitud máxima	Único	Requerido
Primer apellido	Solo caracteres [A-Z], sin acentos ni "ñ"	20	No	Sí
Segundo apellido	Solo caracteres [A-Z], sin acentos ni "ñ"	20	No	Sí
Primer nombre	Solo caracteres [A-Z], sin acentos ni "ñ"	20	No	Sí
Otros nombres	Solo caracteres de la A a la Z mayúscula, sin acentos ni "ñ". Permite el carácter espacio entre nombres.	50	No	No
País de empleo	Solo puede ser Colombia o Estados Unidos	N/A	No	Sí
Tipo de identificación	Solo puede ser cédula de ciudadanía, cédula de extranjería, pasaporte o permiso especial.	N/A	No	Sí
Número de identificación	Solo caracteres [A-Z], [a-z], [0-9] y -.	20	Sí	Sí
Se debe generar automáticamente con el primer nombre y primer apellido del empleado de la siguiente forma:				
Correo electrónico	<PRIMER_NOMBRE>.<PRIMER_APELLIDO>.<ID>@<DOMINIO>	300	Sí	Sí
Si llegase a existir otro correo igual, debe agregarse un número de identificación después del primer apellido. El dominio es creado según el país de empleo.				
Fecha de ingreso	Solo permite fechas comprendidas entre 30 días antes a la fecha actual hasta un día antes a la fecha actual.	N/A	No	Sí
Área	Solo puede ser Administración, Financiera, Compras, Infraestructura, Operación, Talento Humano, Servicios Varios u Otro.	N/A	No	Sí
Estado	Solo puede ser Activo.	N/A	No	Sí
Fecha y hora de registro	Debe estar en el siguiente formato DD/MM/YYYY HH:mm:ss	N/A	No	Sí

Figura 2.3: notas adicionales del requisito EM-01.

2.4.2 Nota 2

Al momento de registrar empleados nuevos se deben cumplir las siguientes condiciones:

- Se debe poder modificar todos los atributos excepto Fecha de Registro.
- Se debe generar un nuevo correo electrónico según las condiciones expuestas.

Capítulo 3

Diseño y Tecnologías de Software

3.1 Arquitectura

Para el diseño del software en un ambiente web se optó por usar API REST en conjunto de una arquitectura por capas por su sencillez y fácil desarrollo. En la siguiente figura se puede observar la arquitectura general de la aplicación web.

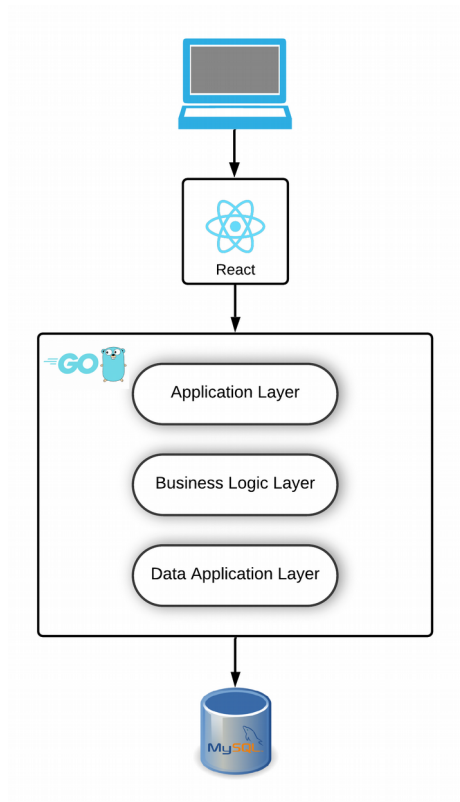


Figura 3.1: arquitectura del software con React, Golang y MySQL.

3.2 Tecnologías de Software

Por restricciones del proyecto se opta por usar Golang en el backend, sin embargo, por elección propia se decide usar React y MySQL en frontend y bases de datos respectivamente.

Capítulo 4

Pruebas de Software

4.1 Pruebas Unitarias

Poder verificar que el software desarrollado está funcionando como se espera es fundamental para lograr un producto de calidad. En general, todas las pruebas fueron diseñadas tomando en cuenta tres casos:

- **Normales.** Situaciones en las que se espera que el software enfrente la mayor parte del tiempo.
- **Límites.** Situaciones donde el software debe lograr ejecutar acciones de manera adecuada sin importar las características de la entrada. Ejemplo, crear un usuario con el máximo de caracteres permitidos.
- **Incorrectos.** Situaciones donde le software debe lograr retornar una respuesta adecuada para entradas incorrectas sin fallar en el proceso. Ejemplo, crear un usuario con campos vacíos.

Go permite realizar pruebas unitarias sin necesidad de instalar software adicional. Usando esta herramienta se realizaron 58 pruebas unitarias para verificar el correcto funcionamiento del software. La siguiente figura muestra un resumen de estas pruebas.

```
=== RUN    TestGetAllUsers
( 1 )
--- PASS: TestGetAllUsers (0.00s)
=== RUN    TestGetUser
( 7 )
--- PASS: TestGetUser (0.03s)
=== RUN    TestUpdateUser
( 24 )
--- PASS: TestUpdateUser (0.27s)
=== RUN    TestCreateUser
( 21 )
--- PASS: TestCreateUser (0.28s)
=== RUN    TestDeleteUser
( 4 )
--- PASS: TestDeleteUser (0.23s)
=== RUN    TestCountCases

<----- Total cases tested: 57 ----->
--- PASS: TestCountCases (0.00s)
PASS
ok      Backend/tests  0.818s
```

Figura 3.1: resumen de las pruebas unitarias con Go.