MÓDULO 6

LOGOTIPO Y MAQUETON

LEIDY LORENA OSORIO GONZALEZ
ID: 614435

PROYECTO FINAL

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
COURSERA
2023

PROYECTO FINAL
ACTIVVIDAD ORGANIZACION

Organización y preparación a) Análisis
1. Elige una temática:
● trabajadores
SEHEAPP Es un aplicación móvil android para mejorar el registro de riesgo para el sistema de gestión de seguridad y salud en el trabajo organizado sistemáticamente en SEHEAPP.

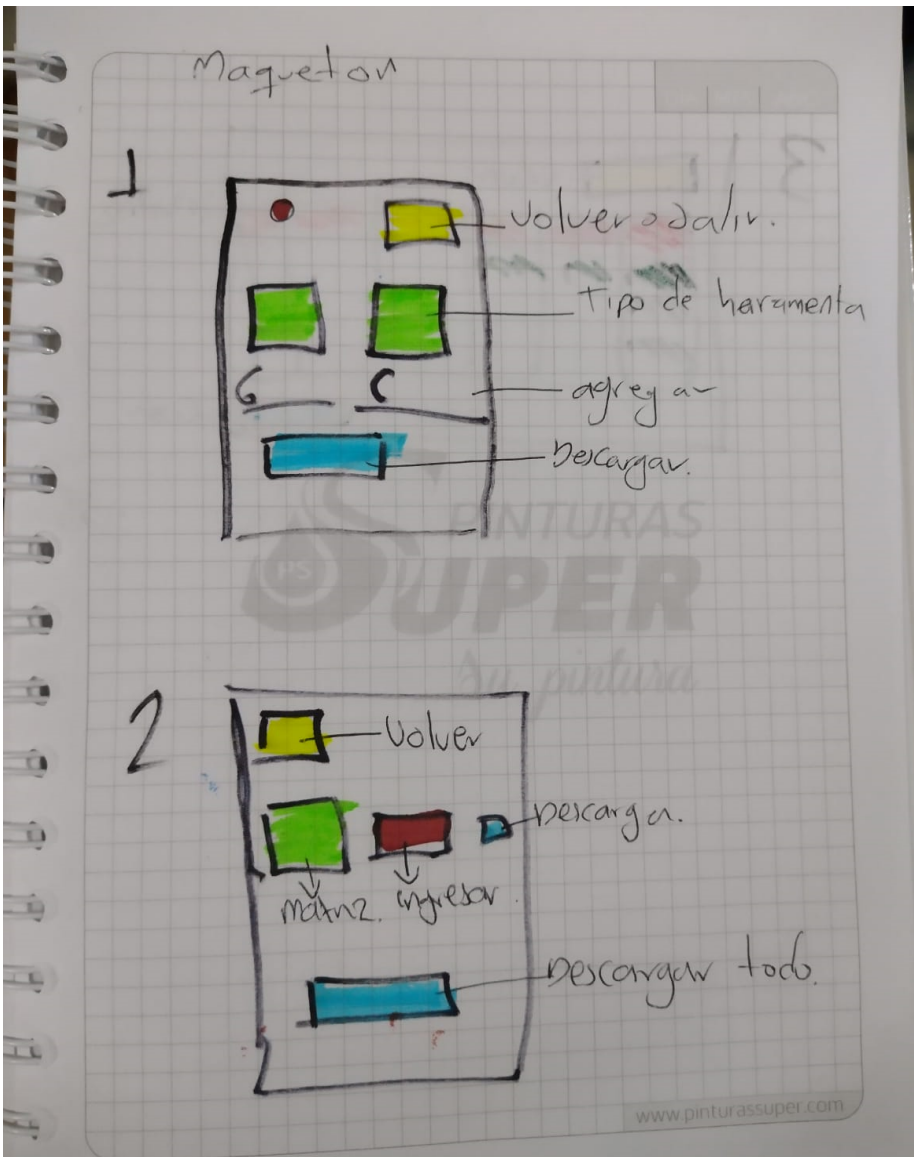
2. ¿Qué problemáticas existen en estos temas para ser resueltas?
las herramientas actuales que se encuentran en la web o app o la común en nuestro ordenador no facilita las actividades de registro por que son de manera manual para garantizar un uso más rápido, adecuado para realizar dichas funciones.

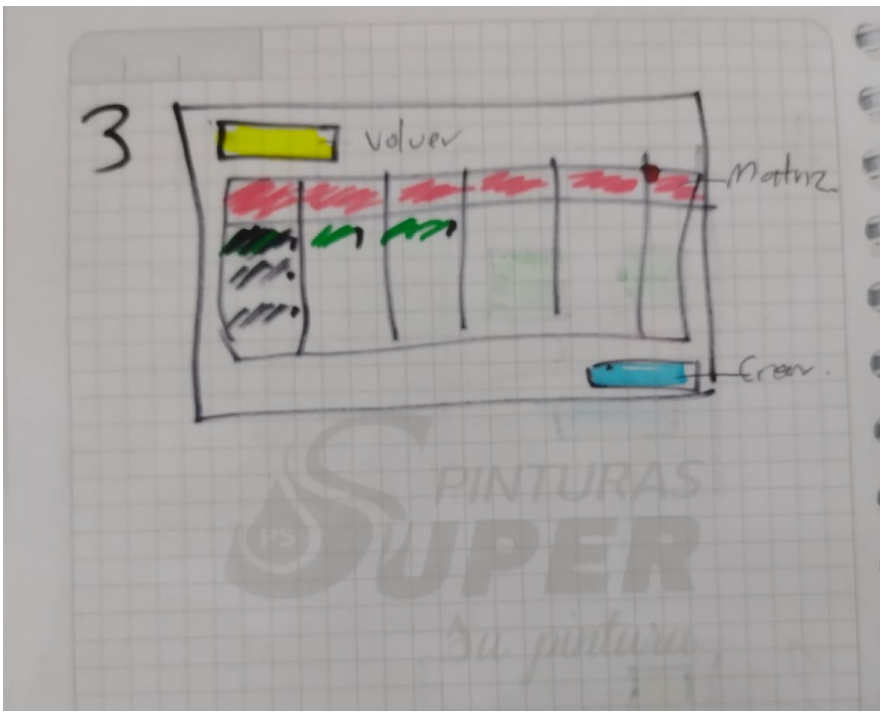
4. ¿Qué necesito para crear mi aplicación? ● Público objetivo: trabajadores del área de Seguridad y salud en el trabajo y/o estudiantes que tengan un aplicativo móvil ● Elementos que pueden ser útiles en el diseño: Material Design : Justinmind ● Elementos que pueden ser útiles en la funcionalidad: lector de código QR. 4. ¿Cómo una aplicación puede contribuir a mejorar estos problemas? la aplicación permite tener herramientas predeterminadas con los valores opcionales y el usuario solo tendría que ponerlos los que necesita, y las app le va dando las opción y respuestas dependiendo la opción y personalización. SEHEAPP optimiza tiempo

5. ¿Cuál es la solución al problema? La app permite que el registro de los riesgos, la evaluación y determinación de riesgos se realice de manera rápida teniendo algunos ítems predeterminados y lo que tendrá que digitar sería poco además de las descripciones básicas

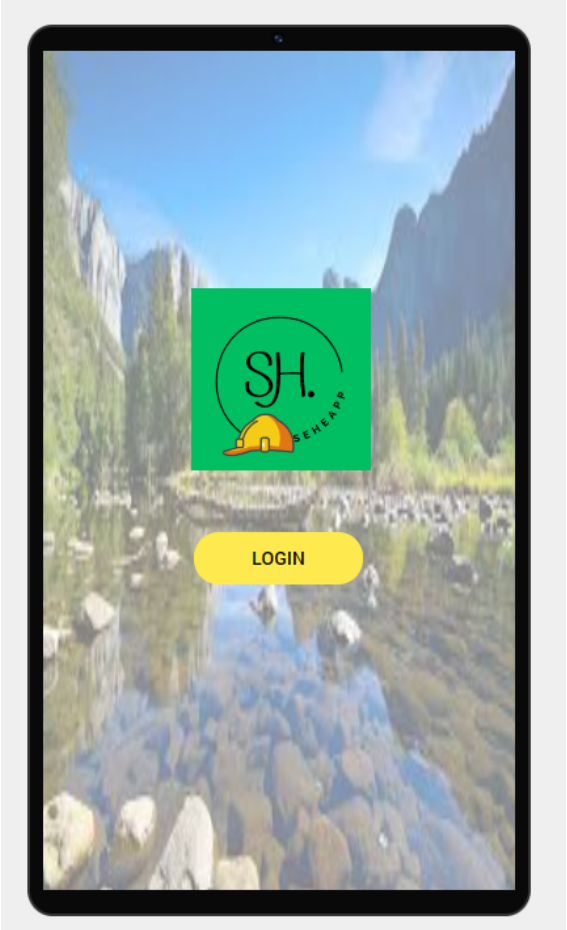
LOGOTIPO APP

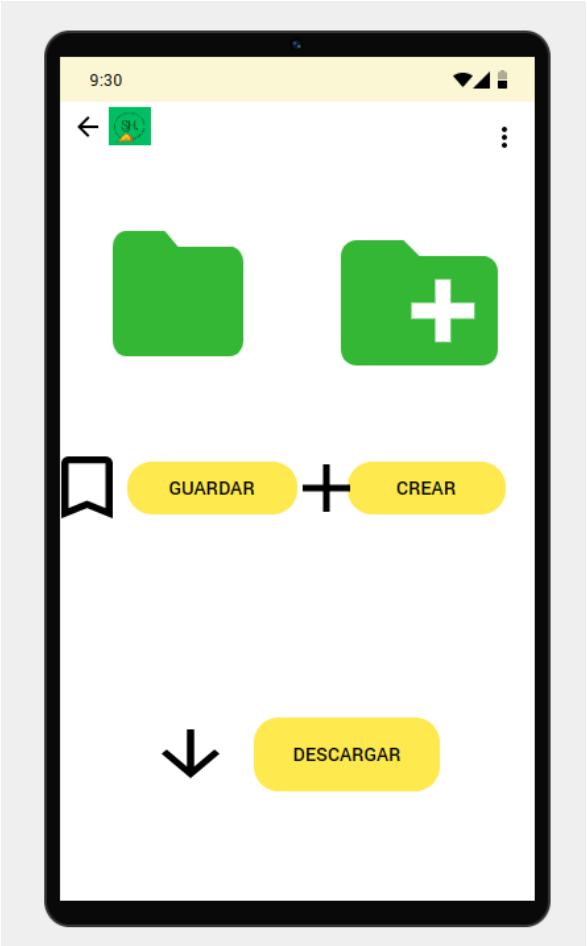
maqueton bosquejo





maqueton software


SH.
SEHEAPP
LOGIN


9:30
GUARDAR
CREAR
DESCARGAR


9:30
Matriz de riesgo
ingresar
descargar
Descargar todo


9:30
Matriz de riesgo
tipo de riesgo
posible riesgo
causa
consecuencia
Matriz de riesgo
N. DEFICIENCIA
VALOR ND
N. EXPOSICION
VALOR DE NE
Matriz de riesgo
N. PROBABILIDAD
N. CONSECUENCIA
N. RIESGO
ACEPTABILIDAD
crear


```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.2"

    defaultConfig {
        applicationId "com.anncode.notificacionfirebase"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
```

```
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.4.0'
    compile 'com.google.firebase:firebase-core:9.0.2'
    compile 'com.google.firebase:firebase-messaging:9.0.2'
    compile 'com.squareup.retrofit2:retrofit:2.0.2'
    compile 'com.squareup.retrofit2:converter-gson:2.0.2'
}

apply plugin: 'com.google.gms.google-services'

mport java.util.Scanner;
public class Automata
{
private String cadena;
private char[] cadenaAutomata;
private int contador;
private boolean aceptado;
public Automata()
{
this.cadena = "00111110";
this.cadenaAutomata = cadena.toChartArray();
this.contador = 0;
this.aceptado = false;
}
public Automata(String cadena)
{
this.cadena = cadena;
this.cadenaAutomata = cadena.toChartArray();
this.contador = 0;
this.aceptado = false;
}
public String getCadena(){return cadena;}
public void setCadena(String cadena)
{
this.cadena = cadena;
this.cadenaAutomata = cadena.toChartArray();
this.contador = 0;
this.aceptado = false;
}
```

```
public void evaluarCadena()
{
this.contador = 0;
this.aceptado = false;
if(!aceptado)
{
System.out.println(cadena+ "La cadena no esta definida en el automata");
}else
{
System.out.println(cadena+ "La cadena sí esta definida en el automata");
}
}
private void evaluarEstadoCero()
{
System.out.println("Estado so ");
if (contador < cadenaAutomata.length)
{
if (cadenaAutomata[contador] == '1')
{
contador++;
evaluarEstadoCero();
}
}else if (cadenaAutomata[contador] == '0'){
contador++;
evaluarEstadoUno();
}
}
private void evaluarEstadoUno()
{
System.out.println("Estado s1 ");
if (contador < cadenaAutomata.length)
{
if (cadenaAutomata[contador] == '1')
{
contador++;
evaluarEstadoUno();
}
}else if (cadenaAutomata[contador] == '0'){
contador++;
evaluarEstadoDos();
}
}
public void evaluarEstadoDos(){
System.out.println("Estado s2 ");
```

```
if (contador < cadenaAutomata.length)
{
while(contador < cadenaAutomata.length)
{
if (cadenaAutomata[contador] == '1')
{
contador++;
} else {
break;
}
}
if( cadenaAutomata[contador]=='0')
{
contador++;
evaluarEstadoTres();
}
}
}
public void evaluarEstadoTres(){
System.out.println("Estado s3 aceptacion");
this.aceptado = true;
if (contador < cadenaAutomata.length)
{
if (cadenaAutomata[contador] == '1')
{
contador++;
evaluarEstadoTres();
} else if(cadenaAutomata[contador] == '0')
{
contador++;
evaluarEstadoError();
}
}
}
private void evaluarEstadoError(){
System.out.println("Estado de error");
this.aceptado = false;
}
public static void main(String[] args)
{
Scanner in = new Scanner(System.in);
System.out.println("Digite Cadena");
String cadena = in.next();
Automata unAutomata = new Automata(cadena);
```

```
unAutomata.evaluarCadena();
}
}
```