

Teste de Integração

1. Verifica a integração do método com o Link e Vídeo no Youtube.

```
public void linkProYoutube(View view) {  
    String url = "https://www.youtube.com/watch?v=HCB2Rxxj7zE";  
    Intent intent = new Intent(Intent.ACTION_VIEW);  
    intent.setData(Uri.parse(url)); // Define a URL correta  
    startActivity(intent);  
}
```

- a. Método do teste:

```
public void testarLinkProYoutube() {  
    View view = new View(this);  
    try {  
        linkProYoutube(view);  
        Log.d("TesteLinkYouTube", "Metodo linkProYoutube ta ok!");  
    } catch (Exception e) {  
        Log.e("TesteLinkYouTube", "Metodo falhou: " + e.getMessage());  
    }  
}
```

- b. Botão chama a função do teste:

XML:

```
<Button android:id="@+id/botaoTestarLinkYouTube"  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
  
    android:text="Testando Link do YouTube" />
```

Java:

```
Button botaoTestarLinkYouTube = findViewById(R.id.botaoTestarLinkYouTube);  
  
botaoTestarLinkYouTube.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        testarLinkProYoutube();  
    }  
});
```

- c. Executar o App, clicar no botão botaoTestarLinkYouTube e acompanhar o LogCat.

1. Verifica a integração do método com o blog informativo.

```
public void SaibaMais(View view) {  
    String url = "https://brasil.un.org/pt-br/sdgs";  
    Intent intent = new Intent(Intent.ACTION_VIEW);  
    intent.setData(Uri.parse(url));  
    startActivity(intent);  
}
```

a. Método do teste:

```
public void testarSaibaMais() {  
    View view = new View(this);  
    try {  
        SaibaMais(view);  
        Log.d("TesteSaibaMais", "Método SaibaMais está tudo ok!");  
    } catch (Exception e) {  
        Log.e("TesteSaibaMais", "Metódo SaibaMais não está funcionando: " + e.getMessage());  
    }  
}
```

b. Inserir testarSaibaMais no Activity correspondente, executar o App e acompanhar o LogCat.

Testes Unitário

1. Verifica a funcionalidade da seleção de dificuldade dos Loops da Função

```
private void carregarQuiz(int dificuldade) {  
    if (quizzesExibidos.size() < 3) {  
        Retrofit retrofit = new Retrofit.Builder()  
            .baseUrl("https://wepapi-fyhrfda9gxdmfmch.canadacentral-01.azurewebsites.net/api/ControllerQuiz/")  
            .addConverterFactory(GsonConverterFactory.create())  
            .build();  
  
        QuizService quizService = retrofit.create(QuizService.class);  
        Call<Quiz> call = quizService.obterQuizAleatorio(String.valueOf(dificuldade));  
  
        call.enqueue(new Callback<Quiz>() {  
            @Override
```

```

        public void onResponse(Call<Quiz> call, Response<Quiz> response) {

            if (response.isSuccessful() && response.body() != null) {

                Quiz quiz = response.body();

                if (quiz.getPerguntas() != null && !quiz.getPerguntas().isEmpty()) {

                    if (!quizzesExibidos.contains(quiz.getId_quiz())) {

                        quizzesExibidos.add(quiz.getId_quiz());

                        configurarPerguntas(quiz);

                    } else {

                        carregarQuiz(dificuldade);

                    }

                } else {

                    Toast.makeText(TelaQuiz5.this, "Não há perguntas suficientes",
                    Toast.LENGTH_SHORT).show();

                }

            } else {

                Toast.makeText(TelaQuiz5.this, "Erro ao carregar o quiz",
                Toast.LENGTH_SHORT).show();

            }

        }

        @Override

        public void onFailure(Call<Quiz> call, Throwable t) {

            Toast.makeText(TelaQuiz5.this, "Falha na conexão: " + t.getMessage(),
            Toast.LENGTH_SHORT).show();

        }

    });

} else {

    Intent intent = new Intent(this, TelaPosQuiz6.class);

    intent.putExtra("resultadosQuizzes", (ArrayList<Map<Integer, Boolean>>)
    resultadosQuizzes);

    startActivity(intent);

}

}

```

a. Método do teste:

```

public void testarCarregarQuiz(int dificuldade) {

```

```

        quizzesExibidos = new ArrayList<>();
        quizzesExibidos.add(1); // IDs fictícios
        quizzesExibidos.add(2);

        carregarQuiz(dificuldade);
        Log.d("TesteCarregarQuiz", "Iniciando CarregarQuiz.");

        if (quizzesExibidos.size() >= 3) {
            Log.d("TesteCarregarQuiz", "Tudo foi carregado corretamente.");
        } else {
            Log.e("TesteCarregarQuiz", "Quantidade total de Quizzes baixa ou não foi possível encontrar");
        }
    }
}

```

2. Verifica a Funcionalidade do botão

```

public void IniciarTelaOpcoes(View view ){
    btnIniciar = findViewById(R.id.btnIniciar);

    Intent intent = new Intent(this, TelaCadastro2.class);
    startActivity(intent);
}

```

a. Método do teste:

```

public void testarIniciarTelaOpcoes() {
    Button btnIniciar = findViewById(R.id.btnIniciar);

    if (btnIniciar != null) {
        btnIniciar.performClick();

        Log.d("Teste", "Intent para TelaCadastro2 funcionou.");
    } else {
        Log.e("Teste", "Botão btnIniciar não foi encontrado.");
    }
}

```

3. Verifica o cadastro de Usuários

```

[HttpPost("Cadastrar")]

public async Task<IActionResult> Cadastrar(Usuario usuario)
{
    if (string.IsNullOrEmpty(usuario.nome))
    {
        return BadRequest("Nome, senha e email são obrigatórios.");
    }
}

```

```

var conn = new MySqlConnection(_connectionString);

{
    await conn.OpenAsync();

    var command = conn.CreateCommand();

    {
        command.CommandText = "INSERT INTO usuario (nome) " +
                                "SELECT @nome " +
                                "WHERE NOT EXISTS (SELECT 1 FROM usuario WHERE nome =
@nome);";

        command.Parameters.AddWithValue("@nome", usuario.nome);

        int rowCount = await command.ExecuteNonQuery();

        if (rowCount > 0)
        {
            return Ok("Usuário adicionado com sucesso.");
        }
        else
        {
            return Conflict("Email já está em uso.");
        }
    }
}
}

```

a. Método do teste

```

public async Task VerificarNomeUsuario(string nomeUsuario, string
resultadoEsperado)
{
    if (string.IsNullOrEmpty(nomeUsuario))
    {

```

```

        Console.WriteLine("O nome do usuário é obrigatório." + resultadoEsperado);
    }
    else
    {
        var conn = new MySqlConnection(_connectionString);
        await conn.OpenAsync();

        var command = conn.CreateCommand();
        command.CommandText = "SELECT COUNT(*) FROM usuario WHERE nome = @Nome";
        command.Parameters.AddWithValue("@Nome", nomeUsuario);

        var result = await command.ExecuteScalarAsync();
        int count = Convert.ToInt32(result);

        if (count > 0)
        {
            Console.WriteLine("Usuário já existe." + resultadoEsperado);
        }
        else
        {
            // Caso o nome não esteja no banco
            Console.WriteLine("Cadastro efetuado." + resultadoEsperado);
        }
    }
}
}

```

b. Casos de Uso

```

await VerificarNomeUsuario("João", "Usuário já existe");

await VerificarNomeUsuario(null, "O nome do usuário é obrigatório");

await VerificarNomeUsuario("", "O nome do usuário é obrigatório");

await VerificarNomeUsuario("Cleber", "Cadastro efetuado");

```

4. Verifica Login do Usuário

```

[HttpPost("Login")]
public async Task<ActionResult> Login([FromBody] UsuarioDto usuario)
{
    if (string.IsNullOrEmpty(usuario.nome))
    {
        return BadRequest(new { sucesso = false, mensagem = "O nome do usuário é obrigatório." });
    }

    var conn = new MySqlConnection(_connectionString);

```

```

{
    await conn.OpenAsync();

    var command = conn.CreateCommand();
    {
        command.CommandText = "SELECT nome FROM usuario WHERE nome =
@Nome";
        command.Parameters.AddWithValue("@Nome", usuario.nome);

        var reader = await command.ExecuteReaderAsync();
        {
            if (await reader.ReadAsync())
            {
                return Ok(new { sucesso = true, mensagem = "Login bem-sucedido." });
            }
            else
            {
                return Unauthorized(new { sucesso = false, mensagem = "Nome de
usuário inválido." });
            }
        }
    }
}
}

```

a. Método do Teste

```

public async Task VerificarLogin(string nomeUsuario, string resultadoEsperado)
{
    if (string.IsNullOrEmpty(nomeUsuario))
    {
        Console.WriteLine("Nome obrigatório para efetuar login" + resultadoEsperado);
    }
    else
    {
        var conn = new MySqlConnection(_connectionString);
        await conn.OpenAsync();

        var command = conn.CreateCommand();
        command.CommandText = "SELECT COUNT(*) FROM usuario WHERE nome =
@Nome";
        command.Parameters.AddWithValue("@Nome", nomeUsuario);

        var result = await command.ExecuteScalarAsync();
        int count = Convert.ToInt32(result);

        if (count > 0)
        {
            Console.WriteLine("Login efetuado" + resultadoEsperado);
        }
    }
}

```

```

    }
    else
    {
        Console.WriteLine("Cadastro necessário" + resultadoEsperado);
    }
}
}

```

b. Casos de Uso

```

await VerificarLogin("João", "Login efetuado");
await VerificarLogin("Cleber", "Cadastro necessário");
await VerificarLogin("", "Nome obrigatório para efetuar login");

```

Teste de Sistema

Caso de Teste	Entrada	Resultado Esperado	Resultado Obtido	Status
Botão Trocar de Pagina	Clique fisico no App	Abrir página de do Intent designidado	Pagina do Intent Abriu corretamente	Aprovado
Botão Sair da Pagina	Clique fisico no App	Abrir página de do Intent designidado	Pagina do Intent Abriu corretamente	Aprovado
Botão iniciar Quis	Clique fisico no App	Abrir página de do Quis	Pagina do Quiz abriu corretamente	Aprovado
Integração com YouTube	Clique fisico na ThumbNail no App	Abrir pagina no App Yputube ou Navegor Padrao e eproduzir video da URL	Abriu o navegadr padrao e reproduziu o video na URL	Aprovado

Integração com Blog	Clique físico no botão SaibaMais do App	Conteúdo do blog exibido no Navegador padrão	Conteúdo exibido corretamente	Aprovado
Segurança/Criptografia	realização do Login	Dados trafegados em Cifra de Cezar	Cifra de Cezar foi aplicada	Aprovado
Login	Login com um Nome de usuário já cadastrado	Login realizado	Login realizado	Aprovado
Cadastro	Cadastro com um nome de usuário ainda não cadastrado	Exibir mensagem "Cadastro concluído"	"Cadastro concluído"	Aprovado
Pergunta do Quiz	Pergunta do Quiz da tabela do Banco de Dados	Pergunta Exibida corretamente	Pergunta Exibida corretamente	Aprovado
Resposta correta do Quiz	Resposta correta selecionada	Resposta correta troca para a cor verde	Resposta correta troca para a cor verde	Falhou
Resposta errada do Quiz	Resposta errada selecionada	Nada acontece	Nada acontece	Aprovado