

```
// Autentica o usuário
var usuario = await _authService.AuthenticateAsync(request.Email, request.Senha);

if (usuario == null)
{
    Console.WriteLine($"[AuthController] Falha na autenticação para: {request.Email}");
    return Unauthorized(new { message = "E-mail ou senha inválidos" });
}

// Gera o token JWT
var token = _authService.GenerateJwtToken(usuario);
var expirationInSeconds = _authService.GetTokenExpirationInSeconds();

  

var response = new LoginResponse
{
    Token = token,
    TokenType = "Bearer",
    ExpiresIn = expirationInSeconds,
    UsuarioId = usuario.Id,
    Nome = usuario.Nome,
    Email = usuario.Email,
    PerfilId = usuario.Perfil
};

Console.WriteLine($"[AuthController] Login bem-sucedido para: {request.Email}");
return Ok(response);
}
catch (Exception ex)
{
    Console.WriteLine($"[AuthController] Erro no login: {ex.Message}");
    return StatusCode(500, new { message = "Erro ao processar login", error = ex.Message });
}
```

Evidência de negação de acesso por falta de autenticação JWT (cód. 401)

The screenshot shows a POST request in the Postman application. The URL is `http://acrwebapprm554826.azurewebsites.net/api/v1/Moto?pageNumber=1&pageSize=10`. The method is set to `POST`, and the URL is displayed in the request field. A `Send` button is visible to the right. Below the request, there are tabs for `Params`, `Authorization`, `Headers (8)`, `Body`, `Scripts`, and `Settings`. The `Params` tab is selected, showing three query parameters: `pageNumber` (value 1) and `pageSize` (value 10). The `Body` tab is also visible. At the bottom, the response status is shown as `401 Unauthorized` with a duration of `6.40 s` and a size of `352 B`. The `Raw` tab is selected in the preview area.