



Universidad ORT Uruguay

Facultad de Ingeniería

Obligatorio - Ingeniería de Software Ágil 2

[Link al repositorio](#)

Justificación de bugs

Integrantes:

Danilo Biladoniga - 231749

Tomás Núñez - 257564

Germán Oller - 242312

Justificación

Para seleccionar los bugs a solucionar nos basamos en la severidad.

Teníamos 2 con severidad crítica (la más alta), por lo que elegimos los issues 7 y 18. El tercer issue fue uno de severidad mayor, el 9, porque consideramos que es el más importante de los de severidad mayor.

Issue 7:

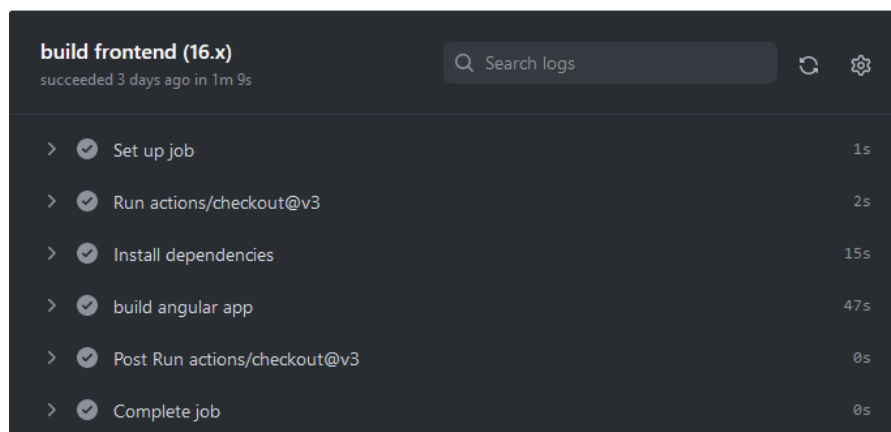
Código de software reparado.

```
<c-header-nav>
  <c-nav-item title="Cart">
    <button class="customButton">
      <a routerLink="/home/cart" cNavLink>
        <svg cIcon name="cil-cart" size="lg"></svg>
      </a>
      <c-badge class="customBadge" color="info">{{ badge }}</c-badge>
    </button>
  </c-nav-item>
  <c-nav-item title="Tracking">
    <a routerLink="/home/tracking" cNavLink>
      <svg cIcon name="cil-compass" size="lg"></svg>
    </a>
  </c-nav-item>
  <c-nav-item title="Login">
    <a routerLink="/login" cNavLink>
      <div>LogIn</div>
    </a>
  </c-nav-item>
  <c-nav-item title="Register">
    <a routerLink="/register" cNavLink>
      <div>Register</div>
    </a>
  </c-nav-item>
</c-header-nav>
```

Código de casos de prueba.

Este issue es un error de frontend, por lo que no tiene pruebas unitarias asociadas. Se realizaron pruebas exploratorias para validar que estuviese solucionado.

Evidencia de ejecución de casos de prueba.



Issue 18:

Código de software reparado.

```
using ExportationModel.Exceptions;
using ExportationModel.ExportDomain;
using ExportationModel.Interfaces;
using Newtonsoft.Json;
using System.Reflection.Metadata;
using System.Xml.Serialization;

namespace XMLExporter
{
    public class XMLFormat : IFormat
    {
        public string GetFormatName()
        {
            return "XML exporter";
        }

        IEnumerable<ExportationModel.ExportDomain.Parameter> IFormat.GetParameters()
        {
            ExportationModel.ExportDomain.Parameter parameter = new ExportationModel.ExportDomain.Parameter()
            {
                InputName = "path",
                InputValue = ""
            };
            return new List<ExportationModel.ExportDomain.Parameter> { parameter };
        }

        public void Export(IEnumerable<DrugExportationModel> drugDtos, IEnumerable<ExportationModel.ExportDomain.Parameter> parameters)
        {
            string path = null;
            foreach (var parameter in parameters)
            {
                if (parameter.InputName == "path")
                {
                    path = parameter.InputValue;
                }
            }
            if (string.IsNullOrEmpty(path))
            {
                throw new InvalidParameterException("The path to export XML file is incorrect.");
            }

            // Serialize the drugDtos to XML
            var serializer = new XmlSerializer(typeof(List<DrugExportationModel>));
            using (var stream = new StreamWriter(path))
            {
                serializer.Serialize(stream, drugDtos.ToList());
            }
        }
    }
}
```

Evidencia de ejecución de casos de prueba.

```
odigo\Backend\PharmaGo.Test\DataAccess.Test\Purchases
DetailRepositoryTest.cs(26,36): warning CS0169: The
field 'PurchasesDetailRepositoryTest.pharmacyReposit
ory' is never used [D:\ORT\8\Agil2\obligatorio-bilad
oniga-nunez-oller\Codigo\Backend\PharmaGo.Test\Pharm
aGo.Test.csproj]
PharmaGo.Test -> D:\ORT\8\Agil2\obligatorio-bilado
niga-nunez-oller\Codigo\Backend\PharmaGo.Test\bin\
Debug\net6.0\PharmaGo.Test.dll
Test run for D:\ORT\8\Agil2\obligatorio-biladoniga-nunez-oller\Codigo\Backend\PharmaGo.Test\bin\Debug\net6.0\PharmaGo.Test.dll (.NETCoreApp,Version=v6.0)
Microsoft (R) Test Execution Command Line Tool Version 17.3.3 (x64)
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.

Passed! - Failed:    0, Passed:   291, Skipped:    0, Total:   291, Duration: 3 s - PharmaGo.Test.dll (net6.0)
```

build backend

succeeded yesterday in 50s

Search logs



- ✓

Test with the dotnet CLI

12s
- ```
375 Warning: /home/runner/work/obligatorio-biladoniga-nunez-oller/obligatorio-biladoniga-nunez-oller/Codigo/Backend/PharmaGo.Test/PharmaGo.Test.csproj]
/home/runner/work/obligatorio-biladoniga-nunez-oller/obligatorio-biladoniga-nunez-oller/Codigo/Backend/PharmaGo.Test/DataAccess.Test
/PurchasesRepositoryTests.cs(29,36): warning CS0169: The field 'PurchasesRepositoryTests.pharmacyRepository' is never used [/home/runner
/work/obligatorio-biladoniga-nunez-oller/obligatorio-biladoniga-nunez-oller/Codigo/Backend/PharmaGo.Test/PharmaGo.Test.csproj]
376 PharmaGo.Test -> /home/runner/work/obligatorio-biladoniga-nunez-oller/obligatorio-biladoniga-nunez-oller/Codigo/Backend/PharmaGo.Test
/bin/Debug/net6.0/PharmaGo.Test.dll
377 Test run for /home/runner/work/obligatorio-biladoniga-nunez-oller/obligatorio-biladoniga-nunez-oller/Codigo/Backend/PharmaGo.Test/bin/Debug
/net6.0/PharmaGo.Test.dll (.NETCoreApp,Version=v6.0)
378 Microsoft (R) Test Execution Command Line Tool Version 17.7.1 (x64)
379 Copyright (c) Microsoft Corporation. All rights reserved.
380
381 Starting test execution, please wait...
382 A total of 1 test files matched the specified pattern.
383
384 Passed! - Failed: 0, Passed: 291, Skipped: 0, Total: 291, Duration: 3 s - PharmaGo.Test.dll (net6.0)
```

0s
- >

✓

Post Setup dotnet

0s
- >

✓

Post Run actions/checkout@v4

0s
- >

✓

Complete job

0s

## Issue 9:

### Código de software reparado.

```
public User CreateUser(string UserName, string UserCode, string Email, string Password, string Address, DateTime RegistrationDate)
{
 string validUserCode = @"^[0-9]{6}$";
 Regex rgUserCode = new(validUserCode);
 string validEmail = @"^[a-zA-Z0-9]{1,30}@[a-zA-Z0-9]{1,30}\.([a-zA-Z]{2,3})+$";
 Regex rgEmail = new(validEmail);
 string validPassword = @"^(?=.*[A-Za-z0-9])(?=.*[\W_]).{8,}$";
 Regex rgPassword = new(validPassword);

 if (String.IsNullOrEmpty(UserName)) throw new InvalidResourceException("Invalid Username");
 if (String.IsNullOrEmpty(UserCode) || !rgUserCode.IsMatch(UserCode)) throw new InvalidResourceException("Invalid UserCode");
 if (String.IsNullOrEmpty(Email) || !rgEmail.IsMatch(Email)) throw new InvalidResourceException("Invalid Email");
 if (String.IsNullOrEmpty>Password || !rgPassword.IsMatch>Password)) throw new InvalidResourceException("Invalid Password");
 if (String.IsNullOrEmpty(Address)) throw new InvalidResourceException("Invalid Address");

 Invitation invitation = _invitationRepository.GetOneDetailByExpression(x => x.UserName.ToLower() == UserName.ToLower() && x.UserCode == UserCode && x.IsActive);
 if (invitation == null)
 throw new ResourceNotFoundException("Invitation not found or is not currently active");
 User exists = _userRepository.GetOneByExpression(u => u.UserName.ToLower() == UserName.ToLower());
 if (exists != null)
 throw new InvalidResourceException("Invalid Username, Username already exists");
 exists = _userRepository.GetOneByExpression(u => u.Email.ToLower() == Email.ToLower());
 if (exists != null)
 throw new InvalidResourceException("Invalid Email, Email already exists");

 User user = new User
 {
 UserName = UserName,
 Email = Email,
 Address = Address,
 Password = Password,
 RegistrationDate = RegistrationDate,
 Pharmacy = invitation.Pharmacy,
 Role = invitation.Role
 };
 _userRepository.InsertOne(user);

 invitation.IsActive = false;
 _invitationRepository.UpdateOne(invitation);

 _userRepository.Save();
 _invitationRepository.Save();

 return user;
}
```

```
string validPassword = @"^(?=.*[A-Za-z0-9])(?=.*[\W_]).{8,}$";
Regex rgPassword = new(validPassword);
```

### Código de casos de prueba.

```
[TestMethod]
public void Create_User_0k()
{
 //Arrange
 var UserName = "pedro901";
 var UserCode = "900357";
 var Address = "Av. Italia 4478";
 var Email = "juan@gmail.com";
 var Password = "Abcdcf12345678.";
 var RegistrationDate = new DateTime(2022, 09, 20, 14, 00, 00);

 _userRepository.Setup(x => x.GetOneByExpression(u => u.UserName.ToLower() == UserName.ToLower())).Returns(exists);
 _userRepository.Setup(x => x.GetOneByExpression(u => u.Email.ToLower() == Email.ToLower())).Returns(exists);

 _invitationRepository
 .Setup(x => x.GetOneDetailByExpression(i => i.UserName.ToLower() == UserName.ToLower() && i.UserCode == UserCode && i.IsActive)).Returns(invitation);
 user.Pharmacy = invitation.Pharmacy;
 user.Role = invitation.Role;

 _userRepository.Setup(x => x.InsertOne(it.IsAny<User>()));
 _userRepository.Setup(x => x.Save());

 invitation.IsActive = false;
 _invitationRepository.Setup(x => x.UpdateOne(invitation));
 _invitationRepository.Setup(x => x.Save());

 //Act
 var response = _userManager.CreateUser(UserName, UserCode, Email, Password, Address, RegistrationDate);

 //Assert
 Assert.IsNotNull(response);
 Assert.AreEqual(response.Pharmacy.Id, 1);
 Assert.AreEqual(response.Role.Id, 1);
}
```

```

[TestMethod]
public void Create_User_Fail_Invalid_Password_Empty()
{
 //Arrange
 var UserName = "pedro901";
 var UserCode = "123456";
 var Address = "Av. Italia 4478";
 var Email = "pedro@gmail.com";
 var Password = "";
 var RegistrationDate = new DateTime(2022, 09, 20, 14, 00, 00);

 //Act
 try
 {
 var response = _userManager.CreateUser(UserName, UserCode, Email, Password, Address, RegistrationDate);
 }
 catch (InvalidResourceException ex)
 {
 //Assert
 Assert.IsNotNull(ex);
 Assert.AreEqual(ex.Message, "Invalid Password");
 }
}

```

Evidencia de ejecución de casos de prueba.

|   |   |                                 |         |
|---|---|---------------------------------|---------|
| ▲ | ✔ | PharmaGo.Test (291)             | 2,7 sec |
| ▶ | ✔ | PharmaGo.Test.BusinessLogic...  | 465 ms  |
| ▶ | ✔ | PharmaGo.Test.DataAccess.Test   | 2 sec   |
| ▶ | ✔ | PharmaGo.Test.WebApi.Test (...) | 188 ms  |