

**DOCUMENTO CON LAS VERIFICACIONES DE CONDICIONES DE CALIDAD DEL PRODUCTO
DE SOFTWARE**

GA11-220501098-AA3-EV02.

(Recopilación de las evidencias GA11-220501098-AA2-EVO1, AA2-EVO2 Y AA3-EVO1)

**LUZ MYRIAM GALEANO QUIROGA
KAREN ESTEFANIA SANABRIA VALERO
LAURA MILENA HENAO GONZALEZ**

**CENTRO DE SERVICIOS FINANCIEROS SENA
ANALISIS Y DESARROLLO DE SOFTWARE**

**MEDELLIN
04 MARZO 2024**

INTRODUCCION

Este informe proporciona un análisis detallado del comportamiento del software AGROAGCA, el cual fue desarrollado, basado en unos requisitos funcionales y no funcionales emitidos por el “cliente”, quien es nuestro objetivo principal para satisfacer sus necesidades y especialmente las de los usuarios del software, es importante aclarar qué para este proyecto estamos usando datos de personas ficticias, ya que este es un proyecto de desarrollo para nuestra formación académica y no un proyecto real; todo este proceso a sido durante el período Octubre 2022 hasta la fecha (Marzo 2024), con el objetivo de evaluar su rendimiento, estabilidad y eficacia en relación con los requisitos y expectativas establecidos.

En cuanto a la Metodología, se utilizó un enfoque integral para evaluar el software, que incluyó pruebas funcionales.

Se recolectaron datos utilizando pruebas automatizadas con Postman, Spring Tool Suite y MySQL para la base de datos y se llevaron a cabo pruebas en entornos de desarrollo para obtener una visión completa del comportamiento del software.

Las buenas prácticas de calidad son una compilación de métodos o técnicas que permiten optimizar las actividades de desarrollo de software.

La documentación técnica adecuada hará que la información sea de fácil acceso y también reducirá la curva de aprendizaje.

El proceso de verificación del software es crucial para garantizar la calidad y el rendimiento adecuados de un producto de software.

En este informe, se presentan las experiencias aprendidas durante el proceso de verificación del software AGROAGCA, destacando las lecciones clave que pueden ser útiles para futuros proyectos y mejoras.

Una organización siempre debe estar en pro de mejorar todos sus procesos, para ser cada vez más competitiva dentro de un mercado que día a día es más exigente, por lo que debe realizar propuestas de mejora que puedan contribuir a que la compañía sea más sobresaliente que otras empresas. Además, toda empresa debe intentar ser cada día mejor que el anterior y para lograrlo, debe sin duda realizar ajustes en cada proceso que de alguna forma impacte positivamente su funcionamiento.

Basado en lo anterior, es importante realizar acciones de mejora, en las cuales se especifiquen las actividades que deben realizarse para alcanzar los objetivos planteados, para ello hay que determinar las tareas, las diferentes acciones a desarrollar, el período de obtención y los indicadores de seguimiento de las mismas.

SELECCIÓN DE BUENAS PRACTICAS DE CALIDAD DE ACUERDO CON EL REFERENTE DE LOS MARCOS DE TRABAJO

Las buenas prácticas de calidad, en el desarrollo de software son fundamentales para garantizar la entrega de productos confiables y satisfactorios para los usuarios.

El marco de trabajo utilizado en el desarrollo de software puede influir en las prácticas de calidad que son más adecuadas o comúnmente recomendadas.

Entonces para iniciar con esta selección, primero proporcionar los requisitos del proyecto, tanto funcionales como no funcionales:

- Se anexa link para acceder a una mejor vista del cuadro de requisitos:

<https://docs.google.com/spreadsheets/d/1aVZsGPQDR724XCF-QpRGAB4kbpXmhdhm/edit?usp=sharing&ouid=112600621916504248799&rtpof=true&sd=true>

IDENTIFICACIÓN				ESTADO				CARACTERÍSTICAS	
Id	Código Jerárquico	Descripción	Tipo	Verión	Estado	Fecha de estado	Responsable	Prioridad	Complejidad
1	1.1	Para inicio de sesión el usuario deberá registrarse con nombres, apellidos, número de celular, tipo de documento número, dirección	Requisito funcional	1	Solicitado	10/23/2022	Usuario	Alto	Alta
	1.2	El sistema le permitirá realizar actualización de datos cuando el usuario lo requiera	Requisito funcional	1	Solicitado	10/23/2022	Usuario	Alto	Moderado
2	2.1	El sistema le exigirá generar un registro general, porque el servicio es para varios usuarios e identificará si ya el usuario tiene o no un registro previo	Requisito funcional	1	Solicitado	10/23/2022	Usuario	Alto	Moderado
	2.2	El sistema le permitirá visualizar el estado de las compras	Requisito funcional	1	Solicitado	10/23/2022	Usuario	Medio	Moderado
3	3.1	El usuario tendrá acceso a la lista de aguacates	Requisito funcional	1	Solicitado	10/23/2022	Usuario	Medio	Moderado
	3.2	El sistema le dará acceso a los vendedores para ver el historial de ventas	Requisito funcional	1	Solicitado	10/23/2022	Usuario	Medio	Moderado
4	4.1	El sistema debe tener la capacidad sostener N cantidad de usuarios por segundo	Requisito no funcional	1	Solicitado	10/23/2022	Creador del software	Alto	Alta
	4.2	Los datos modificados deben ser actualizados para todos los usuarios en menos de 5 segundos	Requisito no funcional	1	Solicitado	10/23/2022	Creador del software	Medio	Moderado
5	5.1	El usuario tendrá acceso al sistema las 24 horas del día, todos los días	Requisito no funcional	1	Solicitado	10/23/2022	Creador del software	Medio	Baja
	5.2	El sistema debe tener manual técnico, bien especificado, consus funcionalidades de software y hardware, terminos y definiciones	Requisito no funcional	1	Solicitado	10/23/2022	creador del software	Alto	Alta
	5.3	se debe contar con un control de versionamiento	Requisito no funcional	1	Solicitado	10/23/2022	creador del software	Medio	Baja
	5.4	El sistema debe tener manual de uso claramente especificado	Requisito no funcional	1	Solicitado	10/23/2022	Creador del software	Alto	Moderado
6	6.1	El sistema no podrá tener fallas con un tiempo mayor a una hora	Requisito no funcional	1	Solicitado	10/23/2022	Creador del software	Medio	Alta
	6.2	El sistema debe tener una disponibilidad de 99,9% en todo momento que el usuario desee ingresar	Requisito no funcional	1	Solicitado	10/23/2022	Creador del software	Alto	Alta

METODOLOGÍA SCRUM

Esta metodología de software, comprende un conjunto de técnicas, métodos y herramientas prácticas que permiten determinar los objetivos generales y específicos del proyecto, teniendo en cuenta los requisitos y necesidades exigidos por el cliente.

Donde trabajaremos de una manera colaborativa, es decir, que nuestro equipo se ira rotando los roles a medida que avance el proyecto, iniciaremos con los roles de:

Miriam Galeano: Product Owner

Karen Sanabria: Team Development

Laura Henao: Scrum Master

Las funciones de cada integrante en su rol serán:

Product Owner: Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y él es responsable del ROI (Return On Investment) del proyecto (entregar un valor superior al dinero invertido). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las prioriza de forma regular.

Scrum Master: Es la persona que lidera al equipo, guiándolo para que cumpla las reglas y procesos de la metodología. Además, gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI (Return On Investment).

Team Development: Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint.

Ahora bien, este enfoque de trabajo ágil, tiene como prioridad garantizar procesos de adaptabilidad a lo largo del desarrollo del proyecto, debido a que, desde el inicio, se desconoce a cabalidad los requerimientos que pueden surgir a lo largo de este. No obstante, se busca facilitar, priorizar y cumplir con la entrega del producto, simplificando procesos, modificando requisitos, poniendo a prueba el progreso e interactuando con el cliente en todas las etapas del mismo.

Scrum se fundamenta en tres tipos de desarrollo, la primera; la etapa de transparencia que hace referencia a eventos de revisión, reuniones, cronogramas e instrumentos de seguimiento y actualización constantes. La segunda; es la etapa de inspección, que se refiere a la supervisión de actividades realizadas por otras áreas y por último; la etapa de adaptación que permite modificar en la marcha algunos procesos con el objetivo de optimizar los estándares de calidad.

De acuerdo con nuestro marco de trabajo que es la Metodología Scrum, un contexto ágil, sabemos que son muy importantes todas sus prácticas, de las cuales usamos todas a menudo; existen estas prácticas de calidad:

- **Integración continua**: Realizar integraciones frecuentes y automatizadas del código para detectar y corregir problemas de calidad de manera temprana.

- **Desarrollo iterativo e incremental:** Desarrollar el software en pequeñas iteraciones, lo que permite obtener retroalimentación rápida y facilita la adaptación a los cambios de requisitos.
- **Colaboración y comunicación constante:** Fomentar la comunicación abierta y la colaboración entre todos los miembros del equipo, así como con los stakeholders, para asegurar que todos estén alineados en los objetivos de calidad.
- **Pruebas unitarias y de aceptación:** Escribir pruebas unitarias automatizadas y pruebas de aceptación del usuario para garantizar que el software cumpla con los requisitos y funcione correctamente.
- **Retrospectivas regulares:** Realizar reuniones periódicas de retrospectiva para identificar oportunidades de mejora en el proceso y en la calidad del producto.

Ante las anteriores prácticas de calidad, sabemos que es importante adaptar estas prácticas a las necesidades específicas del proyecto y del equipo de desarrollo, teniendo en cuenta las características y los requisitos del software a desarrollar.

ARQUITECTURA

Arquitectura MVC

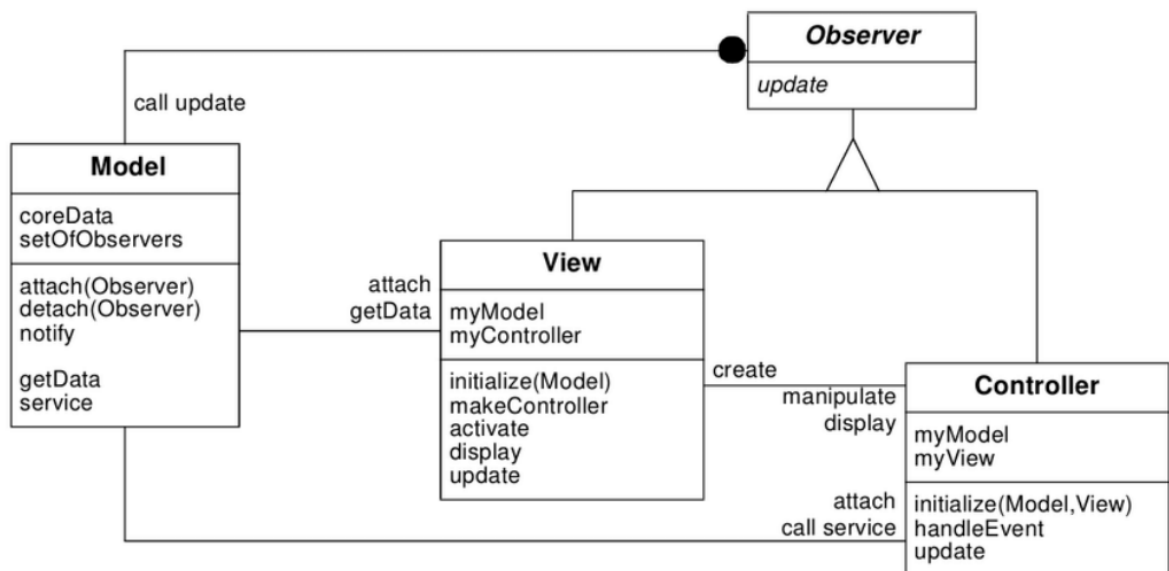


Gráfico 7 Modelo MVC

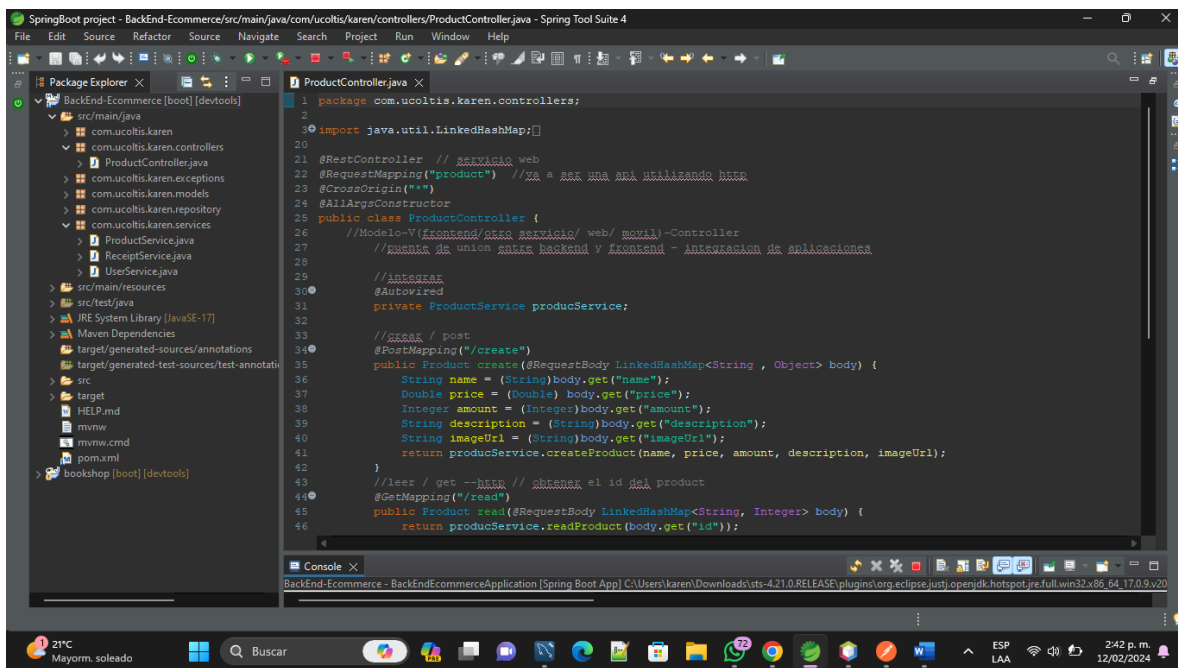
Herramienta de Pruebas de Software.

Pruebas con **POSTMAN**.

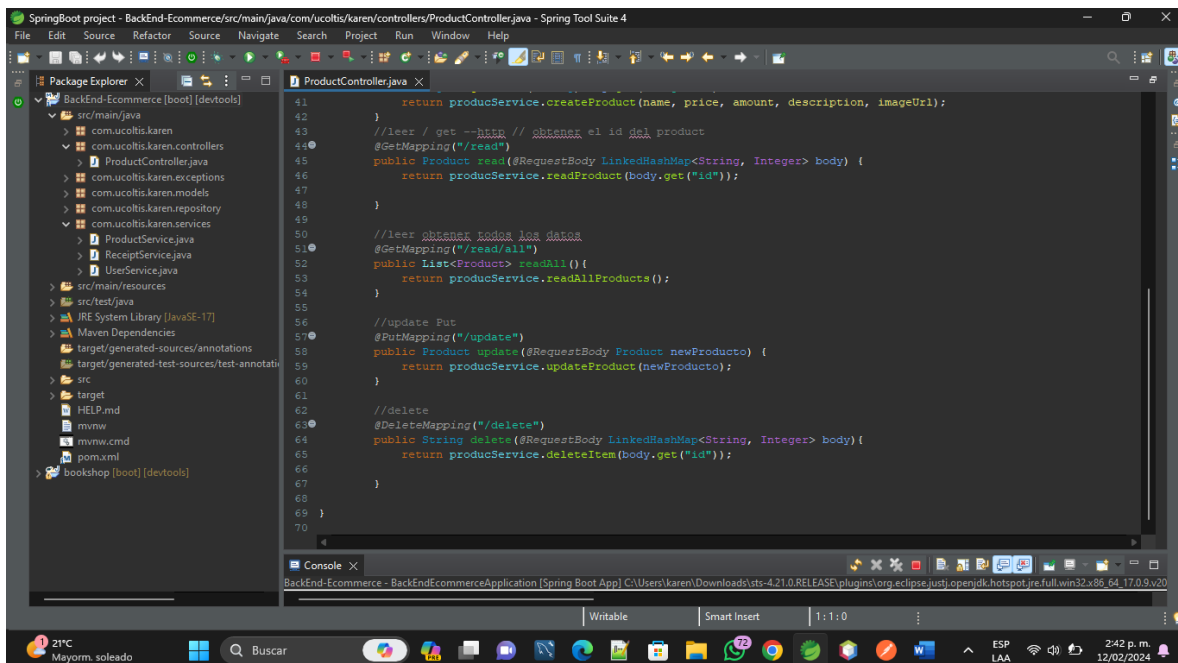
Realizar las pruebas básicas de su solución de software, tomar capturas del proceso y anexo al documento.

BACKEND

Pruebas al servicio producto



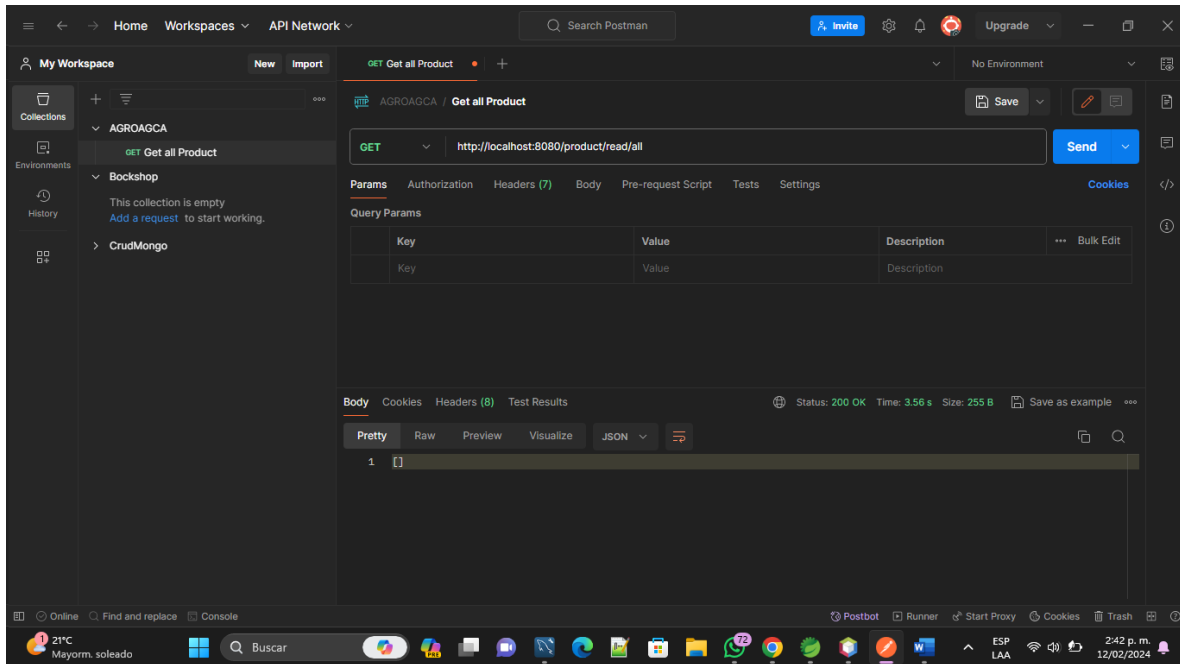
```
1 package com.ucoltis.karen.controllers;
2
3 import java.util.HashMap;
4
5 @RestController // RestAPI web
6 @RequestMapping("product") //Ya = API para manejar datos
7 @CrossOrigin("")
8 @AllArgsConstructor
9 public class ProductController {
10     //Modelo-V(frontend/para servicio/ web/ movil)-Controller
11     //puente de union entre backend y frontend - Integracion de aplicaciones
12
13     //Inyectar
14     @Autowired
15     private ProductService productService;
16
17     //crear / post
18     @PostMapping("/create")
19     public Product create(@RequestBody HashMap<String, Object> body) {
20         String name = (String)body.get("name");
21         Double price = (Double)body.get("price");
22         Integer amount = (Integer)body.get("amount");
23         String description = (String)body.get("description");
24         String imageUrl = (String)body.get("imageUrl");
25         return productService.createProduct(name, price, amount, description, imageUrl);
26     }
27
28     //Leer / get --http // obtener el id del product
29     @GetMapping("/read")
30     public Product read(@RequestBody HashMap<String, Integer> body) {
31         return productService.readProduct(body.get("id"));
32     }
33 }
```



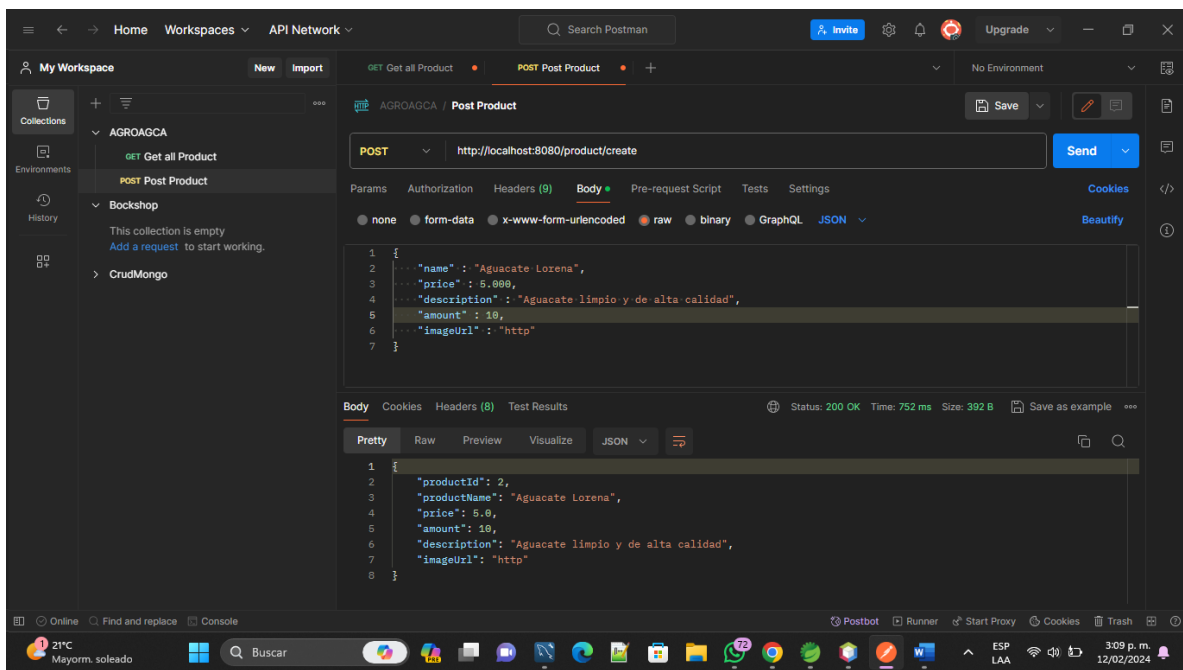
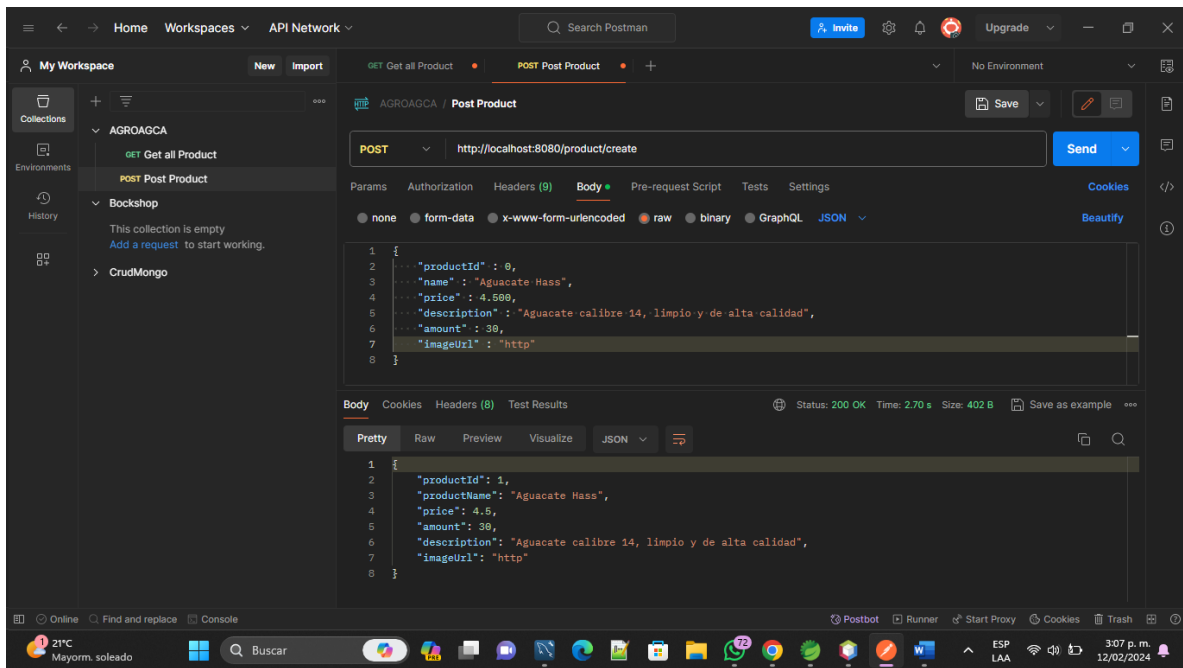
```
41 return productService.createProduct(name, price, amount, description, imageUrl);
42 }
43 //Leer / get --http // obtener el id del product
44 @GetMapping("/read")
45 public Product read(@RequestBody HashMap<String, Integer> body) {
46     return productService.readProduct(body.get("id"));
47 }
48
49 //Leer obtener todas las cosas
50 @GetMapping("/read/all")
51 public List<Product> readAll() {
52     return productService.readAllProducts();
53 }
54
55 //update Put
56 @PutMapping("/update")
57 public Product update(@RequestBody Product newProduct) {
58     return productService.updateProduct(newProduct);
59 }
60
61 //delete
62 @DeleteMapping("/delete")
63 public String delete(@RequestBody HashMap<String, Integer> body) {
64     return productService.deleteItem(body.get("id"));
65 }
66
67 }
68
69 }
70 }
```

(Pruebas)

Get all Product



Post product



Verificamos en la base de datos

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHMAS

bookshop
contactos
control_estudiantes
curso.springboot
ecommerce
hibernate_sequences
items_on_receipt
products
receipt
users
Views
Stored Procedures
Functions
estudiante_db
facturacion

Administration Schemas

Information

Table: products

Columns:

- product_id int PK
- amount int
- description varchar(255)
- image_url varchar(255)
- price double
- produc_name varchar(255)

Object Info Session

1 • SELECT * FROM ecommerce.products;

Result Grid

product_id	amount	description	image_url	price	produc_name
1	30	Aguate calibre 14, limpio y de alta calidad	http	4.5	Aguate Hass
2	10	Aguate limpio y de alta calidad	http	5	Aguate Lorena

products 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:09:50	SELECT * FROM ecommerce.items_on_receipt LIMIT 0, 1000	0 row(s) returned	0.032 sec / 0.000 sec
2	15:09:58	SELECT * FROM ecommerce.products LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

21°C
Mayorm. soleado

Buscar

ESP LAA

3:10 p. m.
12/02/2024

Update Product

Home Workspaces API Network

Search Postman

My Workspace

New Import

GET Get all Product

POST Post Product

PUT Update Product

AGROAGCA / Update Product

PUT http://localhost:8080/product/update

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "productid": 2,
3   "name": "Aguate Lorena",
4   "price": 4.35,
5   "description": "Fresco y de calidad",
6   "amount": 450,
7   "imageurl": "http"
8 }

```

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 1321 ms Size: 367 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "productid": 2,
3   "productName": null,
4   "price": 4.35,
5   "amount": 450,
6   "description": "Fresco y de calidad",
7   "imageurl": "http"
8 }

```

737609
-1.02%

Buscar

ESP LAA

3:16 p. m.
12/02/2024

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHMAS

bookshop
contactos
control_estudiantes
cursoSpringboot
ecommerce
hibernate_sequences
items_on_receipt
products
receipt
users
Views
Stored Procedures
Functions
estudiante_db
facturacion

Administration Schemas

Information

Table: products

Columns:

- product_id int PK
- amount int
- description varchar(255)
- image_url varchar(255)
- price double
- produc_name varchar(255)

Object Info Session

File 9* SQL File 8* SQL File 10* SQL File 10* SQL File 16* Query 1 SQL File 18* SQL File 19* SQL File 15* items_on_receipt products

Limit to 1000 rows

1 SELECT * FROM ecommerce.products;

Result Grid

product_id	amount	description	image_url	price	produc_name
1	30	Aguaate calibre 14, limpio y de alta calidad	http	4.5	Aguaate Hass
2	450	Fresco y de calidad	http	4.35	Aguaate Hass

products 2 x

Apply Revert

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:09:50	SELECT * FROM ecommerce.items_on_receipt LIMIT 0, 1000	0 row(s) returned	0.032 sec / 0.000 sec
2	15:09:58	SELECT * FROM ecommerce.products LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
3	15:17:01	SELECT * FROM ecommerce.products LIMIT 0, 1000	2 row(s) returned	0.016 sec / 0.000 sec

737809 -1.02%

Buscar

ESP LAA

3:17 p. m. 12/02/2024

Pruebas al servicio usuarios

Create user

Home Workspaces API Network

Search Postman

Invite Upgrade

My Workspace

New Import

GET Get all Proc POST Post Prod PUT Update Proc AGROAGCA DEL Delete Proc POST Post Use

No Environment

AGROAGCA

POST Post Users

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "firstName": "Karen",
3   "lastName": "Sanabria",
4   "email": "kazens2@gmail.com",
5   "password": "sv2004",
6   "address": "Bogota",
7   "phoneNumber": "3204838668"
8 }
9

```

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 1833 ms Size: 397 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "userId": 52,
3   "firstName": "Karen",
4   "lastName": "Sanabria",
5   "email": "kazens2@gmail.com",
6   "password": "sv2004",
7   "address": null,
8   "phoneNumber": "3204838668"
9 }

```

Online Find and replace Console

Postbot Runner Start Proxy Cookies Trash

15°C Prac. despejado

Buscar

ESP LAA

8:35 p. m. 12/02/2024

Postman interface showing a PUT request to `http://localhost:8080/users/update`. The request body is a JSON object:

```
{  "userId": 53,  "firstName": "Miryam",  "lastName": "Galeano",  "email": "MiliGaleano@gmail.com",  "password": "galeano34",  "address": "Cali",  "phoneNumber": "3234567898"}
```

The response status is 200 OK, Time: 1533 ms, Size: 406 B. The response body is the same JSON object.

MySQL Workbench interface showing the results of a query:

```
SELECT * FROM ecommerce.users;
```

The result grid shows the following data:

user_id	address	email	first_name	last_name	password	phone_number	address
52		karens2@gmail.com	Karen	Sanabria	sv2004	3204838668	
53		lauh2@gmail.com	Laura	Henao	henao10	3102398087	

The Action Output shows the following actions:

#	Time	Action	Message	Duration / Fetch
1	15:09:50	SELECT * FROM ecommerce.items_on_receipt LIMIT 0, 1000	0 row(s) returned	0.032 sec / 0.000 sec
2	15:09:58	SELECT * FROM ecommerce.products LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
3	15:17:01	SELECT * FROM ecommerce.products LIMIT 0, 1000	2 row(s) returned	0.016 sec / 0.000 sec
4	20:37:15	SELECT * FROM ecommerce.users LIMIT 0, 1000	2 row(s) returned	0.015 sec / 0.000 sec

Update user

The screenshot shows the Postman application interface. On the left, the 'Collections' pane is expanded to 'AGROAGCA', which contains several endpoints. The 'PUT Update Users' endpoint is selected. The main workspace shows the request configuration for a PUT request to 'http://localhost:8080/users/update'. The 'Body' tab is active, displaying a JSON payload for a user update. The status bar at the bottom indicates the request was successful with a 200 OK status.

Request Configuration:

- Method: PUT
- URL: http://localhost:8080/users/update
- Body Type: raw

Request Body (JSON):

```
1 {
2   "userId": 53,
3   "firstName": "Miryam",
4   "lastName": "Galeano",
5   "email": "MiliGaleano@gmail.com",
6   "password": "galeano34",
7   "address": "Cali",
8   "phoneNumber": "3234567898"
9 }
```

Response: Status: 200 OK, Time: 1533 ms, Size: 406 B

The screenshot shows the MySQL Workbench application interface. The 'Schemas' pane on the left lists the database structure, including the 'ecommerce' database and its tables. The 'Query' editor is open, showing a SQL query to select all records from the 'ecommerce.users' table. The 'Result Grid' displays the query results, showing two rows of user data. The 'Action Output' pane at the bottom shows the execution log of the query.

Query:

```
1 SELECT * FROM ecommerce.users;
```

Result Grid:

user_id	address	email	first_name	last_name	password	phone_number	address
52		karens2@gmail.com	Karen	Sanabria	sv2004	3204838668	
53		MiliGaleano@gmail.com	Miryam	Galeano	galeano34	3234567898	Cali

Action Output:

#	Time	Action	Message	Duration / Fetch
1	15:09:50	SELECT * FROM ecommerce.items_on_receipt LIMIT 0, 1000	0 row(s) returned	0.032 sec / 0.000 sec
2	15:09:58	SELECT * FROM ecommerce.products LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
3	15:17:01	SELECT * FROM ecommerce.products LIMIT 0, 1000	2 row(s) returned	0.016 sec / 0.000 sec
4	20:37:15	SELECT * FROM ecommerce.users LIMIT 0, 1000	2 row(s) returned	0.015 sec / 0.000 sec
5	20:48:38	SELECT * FROM ecommerce.users LIMIT 0, 1000	2 row(s) returned	0.032 sec / 0.000 sec

Login user

Postman interface showing a POST request to `http://localhost:8080/users/login`. The request body is a JSON object:

```
1 {
2   "email": "MiliGaleano@gmail.com",
3   "password": "galeano34"
4 }
```

The response is a JSON object:

```
1 {
2   "userId": 53,
3   "firstName": "Miryam",
4   "lastName": "Galeano",
5   "email": "MiliGaleano@gmail.com",
6   "password": "galeano34",
7   "address": "Cali",
8   "phoneNumber": "3234567898"
9 }
```

Status: 200 OK, Time: 1519 ms, Size: 406 B

Read User

Postman interface showing a GET request to `http://localhost:8080/users/read?id=52`. The response is a JSON object:

```
1 {
2   "userId": 52,
3   "firstName": "Karen",
4   "lastName": "Sanabria",
5   "email": "karens2@gmail.com",
6   "password": "sv2084",
7   "address": null,
8   "phoneNumber": "3264838668"
9 }
```

Status: 200 OK, Time: 471 ms, Size: 397 B

FRONTEND

Nueva pestaña x registroCliente.html x +

Archivo C:/Users/karen/Downloads/AGROAGCA%20AGC%20Front/AGROAGCA%20AGC%20Front/agc/AGC/cliente/registro/registroCliente.html

Gmail YouTube Maps Noticias Traducir Ser...

Esta página dice
Por favor, completa todos los campos.

Aceptar

Nombre Apellido

Tipo de persona:
☒ Jurídica ☐ Natural

Dirección

Correo electrónico Número de teléfono

Nit Número de documento

Contraseña Confirmar Contraseña*

Siguiente

20°C Mayorm. nublado

Buscar

ESP LAA

2:37 p.m. 14/02/2024

Nueva pestaña x Agroagca : Asociación de Aguas x +

Archivo C:/Users/karen/Downloads/AGROAGCA%20AGC%20Front/AGROAGCA%20AGC%20Front/agc/AGC/cliente/login/LoginCliente.html

Gmail YouTube Maps Noticias Traducir Serie HP Smart Tank... Codificación de alg... Nueva pestaña

Agroagca
ASOCIACIÓN DE AGUAS

Iniciar sesión Cliente

Incluye un signo "@" en la dirección de correo electrónico. La dirección "karensanabria" no incluye el signo "@".

Contraseña

Acceder

[Olvidaste tu contraseña?](#)
[Aún no tienes cuenta? Regístrate aquí](#)

20°C Mayorm. nublado

Buscar

ESP LAA

2:35 p.m. 14/02/2024

Nueva pestaña x Formulario de Compra x +

Archivo C:/Users/karen/Downloads/AGROAGCA%20AGC%20Front/AGROAGCA%20AGC%20Front/agc/AGC/cliente/Compra/FormularioCompra.html ☆ 🔍 📁 📄 ⋮

Gmail YouTube Maps Noticias Traducir Serie HP Smart Tank... Codificación de alg... Nueva pestaña | Todos los favoritos

Dirección de Envío

Bogota

Teléfono

3204838668

Correo Electrónico

sanabriakaren029@gmail.com

Medio de Pago

Número de Tarjeta (16 dígitos)

.....

Mostrar/Ocultar

Fecha de Vencimiento

julio de 2024

Haz coincidir el formato solicitado.

CVV (3 dígitos)

Mostrar/Ocultar

Realizar Compra

20°C Mayorm. nublado 2:38 p. m. 14/02/2024

Nueva pestaña x Formulario de Compra x +

Archivo C:/Users/karen/Downloads/AGROAGCA%20AGC%20Front/AGROAGCA%20AGC%20Front/agc/AGC/cliente/Compra/FormularioCompra.html ☆ 🔍 📁 📄 ⋮

Gmail YouTube Maps Noticias Traducir Serie HP Smart Tank... Codificación de alg... Nueva pestaña | Todos los favoritos

Dirección de Envío

Bogota

Teléfono

3204838668

Correo Electrónico

sanabriakaren029@gmail.com

Medio de Pago

Número de Tarjeta (16 dígitos)

.....

Mostrar/Ocultar

Fecha de Vencimiento

julio de 2024

CVV (3 dígitos)

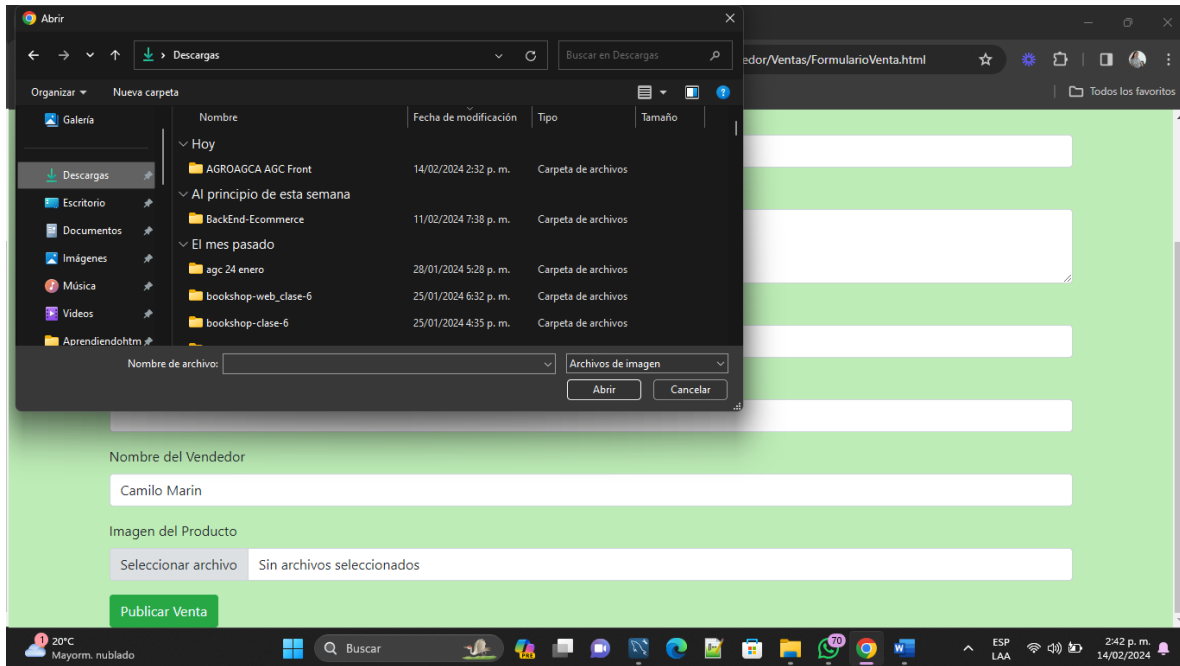
..

Mostrar/Ocultar

Realizar Compra

Haz coincidir el formato solicitado.

20°C Mayorm. nublado 2:38 p. m. 14/02/2024



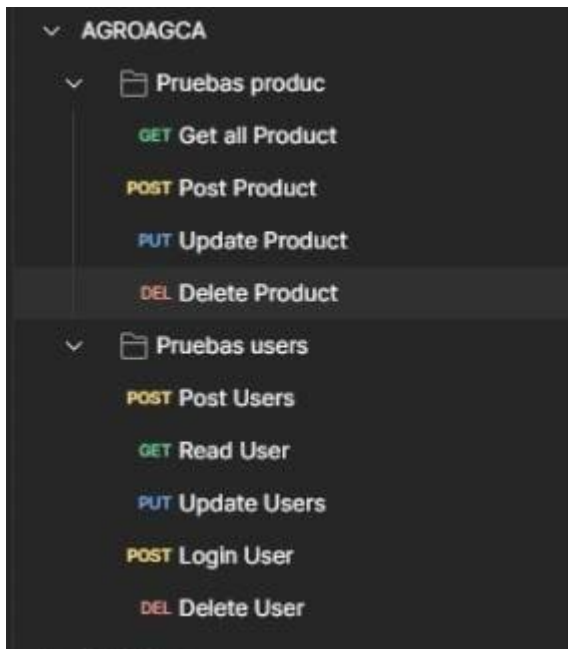
Elaborar un resumen de las pruebas de software realizadas.

Nuestro proyecto es un E-commerce de compraventa de aguacates, donde usamos como ID de desarrollo Spring Tool Suite 4 y dentro de él usando Spring Boot, para la base de datos relacional usamos MySQL y para las pruebas usamos Postman.

Para realizar las pruebas, por supuesto usamos nuestro proyecto, con sus respectivos paquetes de controladores, excepciones, modelos, repositorios y los servicios.

Para iniciar, pusimos a correr nuestra aplicación con Spring Boot App, la cual corrió correctamente.

De ahí vamos a Postman, donde con anticipación habíamos realizado pruebas y están las carpetas creadas.



Iniciando con las **pruebas de los productos**:

GET – Leer el producto (Geta ll product)

POST – Añadir el producto (post product)

PUT – Editar el producto (update producto)

Seguimos con las **pruebas de Usuarios**:

POST – Agregar usuarios (post users)

GET – Leer usuarios (read users)

PUT – Editar usuarios (update users)

POST – verificar usuario y contraseña (login users)



Evidenciando todo lo anterior, damos continuidad a este informe con un registro detallado:

Utilidades o Herramientas para Implementar las Pruebas.

La realización de las pruebas fue realizada con Potsman.
Spring Tool Suite 4 y dentro de él usando Spring Boot.



Para la base de datos relacional usamos MySQL.

Riesgos y Contingencias.

- No tener suficientes miembros en el equipo de pruebas.
- No tener entornos de prueba adecuados para simular el entorno de producción.
- Los requisitos del proyecto cambian durante el desarrollo o las pruebas.
- Las herramientas de pruebas seleccionadas pueden fallar o no ser eficaces.
- Dificultades al integrar diferentes componentes o módulos.
- No disponer de suficiente tiempo para realizar pruebas exhaustivas.

Equipo Evaluador.

El equipo evaluador estuvo compuesto por:

- Laura Milena Henao Gonzalez
- Luz Miriam Galeano Quiroga
- Karen Estefanía Sanabria Valero

Quienes durante todo el proceso y desarrollo fuimos responsables de realizar pruebas, analizar datos y proporcionar retroalimentación sobre el software.

Métricas Utilizadas.

Rendimiento:

El software mostró un rendimiento general sólido durante el período de evaluación.

Se observaron tiempos de respuesta aceptables para la mayoría de las acciones del usuario.

Estabilidad:

No se identificaron fallos durante el período de evaluación.

La mayoría de los fallos fueron relacionados con el registro de los usuarios.

Se tomaron medidas correctivas para abordar los fallos identificados, lo que resultó en una mejora significativa en la estabilidad y rendimiento del software.

Funcionalidad:

Se verificaron todas las características principales del software y se encontraron en funcionamiento según lo previsto.

Se identificó un problema funcional menor, que fueron abordados mediante revisión de códigos, correcciones y mejoras incrementales.

Seguridad:

Se realizaron pruebas de seguridad para identificar vulnerabilidades potenciales.

Se encontraron 3 vulnerabilidades, que fueron clasificadas según su gravedad y se tomaron medidas correctivas para mitigar los riesgos asociados.

Ponderación.

En el contexto de un informe de resultados de comportamiento de un software, la ponderación se utiliza para indicar cuánto valor se le da a cada métrica en comparación con otras en términos de su contribución al rendimiento general del software.

Se debe dar un valor en porcentaje a cada métrica utilizada, estos valores deben sumar un total del 100%.

La ponderación ayuda a dar una visión clara de qué aspectos son considerados más críticos en la evaluación y permite a los evaluadores enfocar sus esfuerzos en áreas prioritarias. Además, proporciona una forma objetiva de comparar diferentes aspectos del software y tomar decisiones informadas sobre cómo abordar áreas de mejora.

Rendimiento: Ponderación del 30%

Estabilidad: Ponderación del 25%

Funcionalidad: Ponderación del 25%

Seguridad: Ponderación del 20%

Fidelidades de Medición.

Se utilizaron herramientas de registros de errores y pruebas automatizadas para garantizar la precisión y consistencia de las mediciones. Se siguieron las mejores prácticas de medición para minimizar el margen de error y garantizar la confiabilidad de los resultados.

Criterios de Aprobación.

Rendimiento: Tiempos de respuesta aceptables para la mayoría de las acciones del usuario, tanto para vendedor como para cliente.

Estabilidad: Número mínimo de fallos y tiempo de inactividad del sistema.

Funcionalidad: Todas las características principales del software funcionan según lo previsto y cumple con nuestras expectativas.

Seguridad: Ausencia de vulnerabilidades críticas identificadas durante las pruebas de seguridad.

Recursos de Infraestructura.

Se utilizaron los siguientes recursos de infraestructura durante la evaluación:

- Entornos de desarrollo, pruebas y producción para simular diferentes condiciones de uso.
- Herramientas de registros de errores para recopilar datos.
- Servidores de prueba para realizar pruebas de carga y evaluación de seguridad.

Conclusiones:

El software AGROAGCA ha demostrado un rendimiento sólido y una funcionalidad confiable durante el período de evaluación. Aunque se identificaron algunos problemas de seguridad, las medidas correctivas implementadas han mejorado significativamente la calidad general del software. Se recomienda continuar monitoreando de cerca el software y realizar pruebas periódicas para garantizar su rendimiento óptimo y seguridad continua.

Recomendaciones:

Realizar pruebas de regresión regularmente para detectar posibles problemas de estabilidad o funcionalidad.

Implementar medidas adicionales de seguridad según sea necesario para mitigar riesgos potenciales.

Continuar recopilando datos de rendimiento y realizando análisis para identificar áreas de mejora.

Este informe refleja el estado actual del software AGROAGCA y proporciona una base para futuras mejoras y desarrollos.

BITÁCORA DE PROCESOS DOCUMENTALES.

Fecha: 4/marzo/2024

Responsable: Laura Henao

Proceso: Informe de Resultados del Comportamiento del Software

Objetivo del Proceso: Registrar resultados de las pruebas del software

Actividades Realizadas

Inicio del Proceso:

- Se revisaron los documentos y herramientas necesarias para iniciar el proceso.
- Se asignaron roles y responsabilidades a los miembros del equipo.
- Hicimos una revisión detallada de las anteriores evidencias para hacer de esta un documento completo.

Recopilación de Información:

- Se recopilaron datos relevantes de evidencias anteriores, realizadas para la generación de las pruebas funcionales de nuestro software AGROAGCA.
- Se verificó la precisión y la integridad de la información recolectada.

Análisis de la Información:

- Se llevó a cabo un análisis detallado de los datos recopilados.
- Cada detalle requerido para este informe fue detallado.

Elaboración de Documentos:

- Se redactaron los documentos necesarios de acuerdo con los hallazgos del análisis.
- Se revisaron los documentos y resultados para garantizar la claridad y la precisión.

Revisión y Aprobación:

- Los documentos fueron revisados por los miembros pertinentes del equipo.
- Los documentos fueron aprobados por los responsables designados.

Conclusiones:

En el anterior informe de resultados del comportamiento del software, se dio a conocer por medio del registro detallado, los diferentes factores solicitados para la creación del mismo como lo son:

- equipo evaluador
- métricas utilizadas
- ponderación
- fidelidades de medición
- criterios de aprobación
- recursos de infraestructura

Firma del responsable: Laura Henao

BITACORAS PARA AGROAGCA

FASE ANALISIS

A través de esta fase se descubre todo lo que se espera del software. Por ello, se realiza una exhaustiva investigación para llegar a una comprensión precisa de los requerimientos o características que debe poseer el programa. De este modo, se elige o crea la arquitectura o estructura en donde operará.

BITACORA DE LOS PROCESOS AGROAGCA			
FASE ANALISIS			
ID	Fecha	Actividad	Observación

1	2/10/2022	Formulación del proyecto de software	Se implementa la idea del proyecto de software a construir durante la formación
2	16/10/2022	Diseño del instrumento de recolección de información	Definición de entrevistas y encuestas como método de recolección de información
3	18/10/2022	Formulario de recolección de información	Realización de los formularios para conocer la necesidad del cliente
4	23/10/2022	Especificación de los requerimientos funcionales y no funcionales	Se hace reconocimiento y documentación de los requerimientos funcionales y no funcionales de acuerdo a los formularios realizados
5	6/11/2022	Identificación de la metodología de desarrollo	Se hace la elección de usar metodología SCRUM para el desarrollo de este proyecto
6	11/12/2022	Elaboración de casos de uso	Se hace el desarrollo de los casos de uso para iniciar el desglose del proyecto
7	13/12/2022	Elaboración historias de usuario	Se hace la creación de historias de usuario historias de usuario
8	5/03/2023	Elaboración de los diagramas de modelo de dominio	Realización de diagramas de modelo de dominio para AGROAGCA
9	12/03/2023	Listas de chequeo para los instrumentos	Se hacen las listas de chequeo para la verificación del proceso realizado hasta el momento
10	26/03/2025	Propuesta técnica y económica	Después de esta recolección de información se presentan dos ofertas al cliente con distintas técnicas y presupuestos para su elección

FASE PLANEACIÓN

Esta primera etapa es esencial porque en ella se determina el ámbito del proyecto, el análisis de los riesgos, el estudio de viabilidad, la duración, la estimación del costo y la asignación de recursos a cada fase. Una planificación bien realizada, permite establecer las bases para un desarrollo orientado al éxito.

BITACORA DE LOS PROCESOS AGROAGCA			
FASE PLANEACIÓN			
ID	Fecha	Actividad	Observación
1	7/05/2023	Identificación de entidades y relaciones planteadas	Se Identifican las entidades y relaciones planteadas
2	10/05/2023	Diseño del modelo conceptual y lógico acorde al caso de estudio	Se diseña el modelo conceptual y lógico acorde al caso de estudio
3	14/05/2023	Programación orientada a objetos	Se hace la investigación sobre POO para el conocimiento y correcto desarrollo del proyecto
4	21/05/2023	Informe de entregables del proyecto de software a construir	Se hace entrega del informe de entregables del proyecto de software
5	22/05/2023	Diagramas de clases	Construcción de diagramas de clase
6	28/05/2023	Arquitectura del software	Se hace la respectiva arquitectura de nuestro software a construir
7	16/06/2023	Construcción de prototipado	Se hace las implementaciones realizadas con técnicas de programación del sistema interactivo propuesto
8	1/07/2023	Maquetación de la interfaz gráfica en HTML	Se inicia la maquetación de la interfaz gráfica que se realizara con HTML y CSS

FASE EJECUCIÓN

La ejecución del proyecto es la etapa del proyecto en la que se pone en marcha todo lo que tu equipo ha planificado. El equipo hace todo lo posible para que los proyectos comiencen con buen pie.

BITACORA DE LOS PROCESOS AGROAGCA			
FASE EJECUCIÓN			
ID	Fecha	Actividad	Observación
1	16/07/2023	Creación de modelo entidad relación	Se realizan para modelar y diseñar bases de datos relacionales, en términos de reglas de negocio y lógicas
2	23/07/2023	Creación de la base de Datos	Se crea la base de datos relacional para AGROGCA en MySQL
3	6/08/2027	Creación del sitio web	Se inicia la elaboración del backend en Java usando SpringToolSuit como id de desarrollo
4	25/08/2023	Diseño del front cumpliendo con los requerimientos	Se realiza la validación del front para verificar que esté cumpliendo con los requerimientos del cliente
5	25/08/2023	Herramientas de versionamiento	Se decide usar GitHub como herramienta de versionamiento
6	28/08/2023	Codificación de módulos	Se inicia la codificación de cada uno de los módulos del software
7	25/09/2023	Verificación de procedimientos para la definición de componentes frontend del sitio web	Se hacen listas de chequeo para la validación de los módulos
8	19/11/2023	Plan de pruebas	Se realiza el plan de pruebas para nuestro proyecto las cuales se harán con postman
9	26/11/2023	Documenta pruebas de software	Se presenta la respectiva documentación de las pruebas de software

FASE EVALUACIÓN

La evaluación del software permite detectar y solucionar posibles problemas en una fase temprana del proceso de desarrollo. Esto puede ayudar a reducir el número de defectos, fallos y errores en el software. Esto puede ahorrar tiempo y dinero a largo plazo y mejorar la calidad general del software.

BITACORA DE LOS PROCESOS AGROAGCA			
FASE EVALUACIÓN			
ID	Fecha	Actividad	Observación
1	11/02/2024	Plan de mantenimiento	Se entrega al cliente el plan de mantenimiento que será requerido por el software
2	12/02/2023	Plan de migración	Se realiza plan de migración para el software en dado caso sea requerido
3	18/02/2024	Documentación técnica	Se hace la elaboración de la documentación técnica del software para ser remitida al cliente
4	18/02/2024	Manual de usuario	Se hace la elaboración del manual de usuario del software para ser remitida al cliente
5	18/02/2024	Plan de capacitación	Se realiza el plan de capacitación para poder hacer entrega del software realizado

Experiencias Aprendidas.

Planificación Detallada:

Aprendizaje: Una planificación detallada es esencial para el éxito del proceso de verificación del software. Pasar tiempo suficiente en la fase de planificación ayuda a identificar los requisitos de verificación, asignar recursos adecuadamente y establecer cronogramas realistas.

La fase del Análisis fue para nosotras un paso fundamental y prohibida omisión ya que es la base del proyecto de software a construir por todo equipo de desarrolladores, porque aquí comprenden los requisitos del sistema, se identifican las necesidades del usuario y se establecen las bases para el diseño y la implementación del software. Durante esta fase, se llevan a cabo actividades como la recopilación de requisitos, la definición de objetivos y alcance del proyecto, y la elaboración de modelos conceptuales del sistema.

La fase de Análisis proporciona una base sólida para el éxito del proyecto de desarrollo de software al garantizar que el equipo comprenda completamente las necesidades y expectativas del usuario, así como las limitaciones y restricciones del proyecto. A través de este proceso, se establecen los cimientos para un diseño y desarrollo efectivos del software.

Acción: En futuros proyectos, dedicaremos más tiempo a la planificación inicial del proceso de verificación, asegurándonos de considerar todos los aspectos relevantes y establecer expectativas claras desde el principio, ya que, para este primer proyecto, no sabíamos a que nos íbamos a enfrentar y en el camino se fueron haciendo cambios de requisitos para ir haciendo del proyecto, un software más funcional o útil, tanto como software como también aprendizaje y conocimiento para nosotras.

Automatización de Pruebas:

Aprendizaje: Este fue un paso bastante complicado al principio, ya que no nos familiarizábamos con las herramientas de pruebas.

La automatización de pruebas puede mejorar significativamente la eficiencia y la cobertura de las pruebas, sabiendo utilizar la herramienta adecuada según las necesidad o prueba a realizar.

Nos dimos cuenta de que las pruebas manuales consumen mucho tiempo y recursos, y pueden dejar margen para errores humanos.

Acción: En el futuro, nos enfocaremos en identificar oportunidades para automatizar las pruebas tanto como sea posible, utilizando herramientas y marcos adecuados para garantizar una cobertura exhaustiva y una ejecución eficiente de las pruebas.

Comunicación, Tiempo y Colaboración:

Aprendizaje: La comunicación clara y la colaboración efectiva entre los equipos de desarrollo y prueba son fundamentales para el éxito del proceso de verificación. La falta de comunicación puede resultar en malentendidos, retrasos y problemas de calidad.

El equipo de desarrollo debe estar en sintonía para realizar cada paso del proyecto ya que de no ser así, se pierden detalles o la coherencia del proyecto.

Acción: En futuros proyectos, fomentaremos una cultura de colaboración, tiempo y comunicación abierta entre los equipos de desarrollo y prueba, facilitando la transferencia de conocimientos y la resolución rápida de problemas.

Desarrollo del Front:

Aprendizaje: El Frontend se refiere a la parte de un sistema de software o aplicación web que los usuarios interactúan directamente en su navegador web u otro dispositivo.

El frontend se encarga de presentar la interfaz de usuario y facilitar la interacción del usuario con el sistema.

Este fue un proceso divertido desde el momento de hacer el maquetado del proyecto, al imaginar cuál sería su apariencia, de ahí empezar a codificar fue una gran experiencia ir viendo cada línea de código como genera un resultado tan vistoso, además aprender a usar librerías que nos facilitan más el diseño es genial.

Acción: Para proyectos futuros, codificar más acciones y poner más funciones novedosas en nuestros proyectos de desarrollo, sacando el máximo provecho de cada uno y obteniendo cada vez resultados más impactantes tanto para nuestra vista como para la de los clientes y usuarios.

Desarrollo del Back:

Aprendizaje: El backend, también conocido como back-end o lado del servidor, se refiere a la parte de un sistema de software o aplicación web que no es directamente accesible para los usuarios finales, pero que se encarga de procesar la lógica de negocio, gestionar la base de datos y proporcionar los datos al frontend para su presentación.

Es un proceso tedioso, estresante, de suma paciencia, análisis y detalle, requiere de mucha concentración al momento de codificar y ver el porque no funciona o conecta una sesión con el servidor y la base de datos, pero al final también se disfrutó la etapa.

La generación de manuales, reportes, informes y bitácoras es ya cuando uno ve la culminación del proyecto.

Acción: Para proyectos futuros, practicar mucho más el backend para ser cada vez mejores y sentirnos como “pez en el agua”, hay que tener cada vez más seguridad para esta.

Conclusiones: El proceso de verificación del software ha proporcionado una valiosa oportunidad para aprender y mejorar nuestras prácticas en el desarrollo de software. Las experiencias aprendidas nos han ayudado a identificar áreas de mejora en la planificación, ejecución, codificación de Front y Back y automatización de pruebas, así como en la comunicación, Tiempo y colaboración entre equipos.

Este informe refleja las experiencias aprendidas durante el proceso de verificación del software AGROAGCA y proporciona una base sólida para futuras mejoras y desarrollos en nuestras prácticas de desarrollo de software.

Firma del responsable del Proyecto:

Laura Henao

Miriam Galeano

Estefanía Sanabria

Documentación de los cambios

Para determinar los Puntos de Función Sin Ajustar, se realiza una consulta en el Documento de Requerimientos presentado anteriormente con el fin de sintetizar los diferentes requisitos para el desarrollo del sistema y clasificarlos según su categoría en esta Métrica de Puntos de Función (Entradas, Salidas, Consultas, Ficheros Internos, Interfaces Externas).

Registro y gestión de usuarios (consultar, crear, modificar y eliminar). Inicio de sesión de múltiples usuarios.

Registro y gestión de clientes y cuentas.

Actualizar y obtener domicilio del cliente.

Ingresar y consultar comentarios en las líneas.

Mensajes de estado al realizar una alteración en las diferentes pantallas (pantallas) de usuario.

CALIDAD BASADA EN PROCESO

Se busca analizar las actividades del proceso que más influyen en la calidad del producto, se modela el proceso para analizarlo mejor y se pueden hacer las siguientes preguntas como:

¿Dónde y cuándo se puede hallar un tipo de defecto?

¿Cómo hallar los defectos antes?

¿Existen alternativas alternas que proporcionen mayor calidad?

MODELOS BASADO EN PROCESO CMM

(Capability Maturity Model) y CMMI (CMM Integrated) Datos Importantes:

Creado como un marco (framework) para varias disciplinas relacionadas:

- Ingeniería de sistemas
- Ingeniería de software
- Desarrollo integrado de productos y procesos Control de proveedores Dos tipos de modelos:

1. Continuo: útil para evaluaciones diferenciadas por proceso y comparaciones detalladas; permite migración de EIA/IS 731 (Industria eléctrica); permite comparación con ISO/IEC 15504
2. Por niveles: útil para comparación agregada; da resultado global que puede compararse con otras empresas; ayuda a migrar desde SWCMM ISO 15504 SPICE (Software Process Improvement and Capability Determination) Datos Importantes:

Ante el aumento de métodos de estimar capacidad y de evaluar procesos, se necesitaba un método más manejable a nivel de proceso, de proyecto. Además, se requiere poder comparar evaluaciones. Nace en Inglaterra aproximadamente en 1995 y se creó como vía alterna a procesos de estandarización. Posee relación con el estándar ISO/IEC 12207.

Propósitos:

Se aplica a mejorar los procesos y a determinar la capacidad aplicable a diferentes dominios, necesidades y tamaño de organización no supone estructura organizacional, filosofía administrativa, modelo e ciclo de vida, tecnologías de software o método de desarrollo usa criterios objetivos y prefiere cuantitativos salida en forma de perfiles comparables Etapas:

- Preparación: alcance, metas del negocio, procesos a evaluar, instancias de proceso
- Recolección de datos: expertos; entrevistas, discusiones, análisis de documentos, herramientas
- Análisis de datos, asignar niveles, preparar salida
- Retroalimentación en resultados

ISO 9000

Datos Importantes:

Estándar en más de 80 países.

ISO 9000-1 a ISO 9000-4 son relativas a Estándares de Administración De Calidad y Aseguramiento de Calidad.

ISO 9000-3: Guía para aplicación de ISO 9001, Desarrollar, Proveer, Instalar y Mantener software para computadora.

ISO 9001: Sistemas de Calidad. Modelo para asegurar calidad en diseño, desarrollo, producción, instalación y servicios de software.

Modelo de proceso: Planear -> Hacer -> Verificar -> Actuar Áreas de procesos:

- Sistema de gestión de calidad
- Responsabilidad de Alta Dirección
- Gestión de recursos
- Realización del producto
- Medición, análisis y mejora Pasos:
- Evaluar necesidades y metas de organización
- Obtener información
- Nombrar consultor
- Toma de conciencia y formación
- Análisis de brecha
- Revisión o definición de procesos
- Suministrar personal
- Establecer cronograma
- Redactar manual de calidad
- Realizar auditoría
- Solicitar certificación
- Realizar evaluaciones

Problemática

El 90% de las empresas desarrolladoras de software son micro y pequeña industria. Las empresas: Son volátiles.

Cuentan con pocos recursos.

Tienen procesos no estandarizados, que dependen del personal que los ejecuta. Luchando por sobrevivir:

Buscan mejorar la calidad de sus productos a través de la mejora de sus procesos.

Estrategias del PROSOFT

Promover exportaciones y la atracción de

inversiones Educación y formación de personal

competente Contar con un marco legal promotor

de la industria Desarrollar el mercado interno

Fortalecer a la industria local

Alcanzar niveles internacionales en capacidad de procesos

Promover la construcción de infraestructura física y de telecomunicaciones CALIDAD BASADA EN PRODUCTO o ISO/IEC 9126 Quality Model

Esta Norma permite especificar y evaluar la calidad del software desde distintas perspectivas, las cuales están asociadas a la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad, y auditoria del software. Puede ser usada por desarrolladores, evaluadores independientes y grupos de aseguramiento de la calidad responsables de especificar y evaluar la calidad del software.

La evaluación de los productos de software que satisfacen las necesidades de calidad del software es uno de los procesos del ciclo de vida de desarrollo del software. La calidad del producto de software puede ser evaluada por medio de la medición de atributos internos, externos o a través de la calidad en uso. El objetivo es que el producto tenga el efecto requerido en un contexto particular de uso.

Métricas:

Funcionalidad

Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.

1. Adecuación
2. Exactitud
3. Interoperabilidad
4. Seguridad
5. Cumplimiento funcional
6. Confiabilidad

Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.

1.	Madurez	4. Cumplimiento de
2.	Recuperabilidad	Fiabilidad
3.	Tolerancia	a fallos
•	Usabilidad	

Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

1.	Aprendizaje	3. Operatividad
2.	Comprensión	4. Atractividad
•	Eficiencia	

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.

Comportamiento en el tiempo

Comportamiento de recursos

Mantenibilidad

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.

1. Estabilidad
2. Facilidad de
3. Facilidad de cambio análisis
4. Facilidad de pruebas

Portabilidad

Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra.

- Capacidad de instalación
- Capacidad

BIBLIOGRAFIA

<https://sena.territorio.la/content/index.php/institucion/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF48/index.html#/curso/tema2>

Evidencias anteriores.