Machine Translated by Google
RCMOS - Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

Planning, execution, and analysis of functional tests in the "Conecta Canaã" system: an approach based on the IEEE 829-2008 standard.

Planning, execution and analysis of functionality tests in the "Conecta Canaã" system: an approach based on IEEE 829-2008

Planning, execution and analysis of functionality tests in the "Conecta Canaã" system: an approach based on the IEEE 829-2008 standard

Ramon Barbosa Pessoa
Federal University of Western Pará
Institute of Engineering and Geosciences
Santarém
Email: 141barbosa@gmail.com

SUMMARY

The digitalization of public services, especially in the context of smart cities, demands impeccable software quality. After all, these systems directly affect the interaction between the administration and citizens. In this work, we tested the functionality of Conecta Canaã, the urban incident management platform of the Municipality of Canaã dos Carajás, in Pará. Following the IEEE 829-2008 standard and the ISO/IEC 25010:2011 quality model, we planned, designed, and executed twelve automated test cases using the Selenium *framework ,* in a technical case study conducted on a real system. We covered three user profiles and seven prioritized functional requirements. All test cases were approved, demonstrating the functional correctness of the evaluated requirements and the conformity of the observed behavior with the specifications, within the limits of the verified scope. These results document a functional testing protocol that can be replicated in other e-government systems in Brazilian municipalities.

Keywords: Software Testing; Software Quality; IEEE 829-2008; Test Automation; Smart Cities.

ABSTRACT

The digitalization of public services, especially in the smart city setting, demands that software quality be impeccable. After all, these systems directly affect the interaction between the administration and citizens. In this work, we tested the functionality of Conecta Canaã, the urban occurrence management platform of the Municipality of Canaã dos Carajás, in Pará. Following the IEEE 829-2008 standard and the ISO/IEC 25010:2011 quality model, we planned, designed and executed twelve automated test cases using the Selenium framework, in a technical case study conducted on a real system. We covered three user profiles and seven prioritized functional requirements. All test cases passed, evidencing the functional correctness of the evaluated requirements and the conformity of the observed behavior with the specifications, within the limits of the verified scope. These results document a functional testing protocol that can be replicated in other e-government systems in Brazilian municipalities.

Keywords: Software Testing; Software Quality; IEEE 829-2008; Test Automation; Smart Cities.

1 INTRODUCTION

The rapid growth of urbanization is among the most defining phenomena of the 21st century and

This places increasing pressure on the need for public services to respond. In Brazil, the rate

Urbanization reached over 84% in 2005, according to estimates from the United Nations.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

They suggest it will reach over 93% by 2050, equivalent to about 237 million people.
concentrated in urban centers (R. Machado, 2023). This demand requires more management models.
efficient and, above all, digitized, capable of creating a relationship between public power and a
an ever-increasing number of citizens.

It is against this backdrop that the smart city model establishes itself as a possible solution.
A city becomes smart when it integrates Information Technologies in a coordinated way.
and Communication to its conventional infrastructure, employing digital resources to expand the
Urban efficiency (Batty et al., 2012). In Brazil, however, scientific production and implementation
Such initiatives are still concentrated in a few regions, revealing a field that is still in its early stages.
and unevenly distributed (Lazzaretti et al., 2019). As a sectoral expression of this paradigm, the
According to Souza et al. (2025), the computerization of public services has been a decisive vector for
Modernization of the State, aligned with the goal of more effective, transparent, and accountable institutions.
as advocated by the 2030 Agenda.

This digitalization materializes in the concept of *smart governance,* understood as the integration between
public administration, citizens and private entities through digital portals and platforms that
They broaden access to public services (R. Machado, 2023). By increasingly recognizing the role
The role of technology in improving internal processes, strengthening transparency, and...
To improve the efficiency of service delivery, government agencies are becoming dependent on software that...
Integrate information, control flows, and optimize resources, increasing data reliability.
and transparency in the use of public goods (Souza et al., 2025). This dependence transfers to the
software takes over responsibility for functions previously performed by human servers, making its
Functional correctness is a requirement as critical as the very existence of the service.

The Conecta Canaã web system, employed, is situated as a concrete instance of this movement.
by the Municipal Government of Canaã dos Carajás (PMCC) to manage urban incidents reported by
citizens and monitored by Internet of Things devices. The municipality, located in the southeast
The state of Pará registered a population growth of 188.73% between 2010 and 2022 — from 26,716 to 77,079.
inhabitants —, which equates to an annual rate of 9.67% (Brazilian Institute of Geography and Statistics).
Statistics, 2022), imposing increasing demands on public services whose supply has not kept pace.
proportionally to that pace. Conecta Canaã is part of a smart city project based on
The Internet of Things (IoT), developed in cooperation between UFPA, UFOPA, and Unifesspa, and accumulates
more than 2,600 incidents handled by the 8 municipal departments since mid-2022, with a rate
with a resolution consistently higher than 65% over the three years of operation (Bacelar et al.,
2025).

For an e-government application to fulfill this role, however, the quality of the software is crucial.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

It is no longer a desirable attribute but a non-negotiable condition. Software quality is understood as...
The systematic conformity of the product with the specified functional and non-functional requirements.
ensured throughout the entire development process (Bartié, 2002; Pressman & Maxim,
2021). The normative reference model adopted in this work to evaluate it is ISO/IEC 25010.
in its 2011 edition (International Organization for Standardization, 2011), which organizes the
Product quality is measured in eight characteristics and broken down into **functional suitability** — the most directly
associated with functional tests — in three sub-characteristics: completeness, correctness and
Appropriability (Estdale & Georgiadou, 2018). Note that the standard was revised in 2023.
(ISO/IEC 25010:2023), which reorganizes the model into nine characteristics; the 2011 edition was
maintained because it was the version in effect at the time of planning and execution of the tests and because it remains unchanged.
The definition of functional adequacy and its three sub-characteristics form the analytical axis of this study.
Figure 1 summarizes the model, highlighting the characteristic and sub-characteristics that guide the analysis.
conducted in this work.

The relevance of this assessment is simultaneously technical, economic, and social. From the very beginning
In commercial systems, the presence of a large number of defects is a constant that affects
usability, functionality and reliability, with direct implications for the business (Rios & Moreira,
2006). The economic impact of this phenomenon is significant: defects detected only in the phase of
Maintenance can cost up to a hundred times more than if corrected during the requirements phase (Fernandes,
(2020), and the report from the *National Institute of Standards and Technology* estimated that the cost of
Defective software was equivalent to about 1% of the US Gross Domestic Product (Rios & Moreira, 2006). In
addition to the economic dimension, there is a social one: in smart government applications, the
low quality directly undermines public trust in digital services (Purbaratri &
(Krisnawaty, 2023). Not by chance, the systematic application of functionality testing is recognized
as a determining factor in software product quality (Andrea, 2007), to the point that the models
maturity levels institutionalize this practice — the *Capability Maturity Model* from level 3 onwards.
(Paulk et al., 1993) and MPS.BR from level B (SOFTEX, 2016).

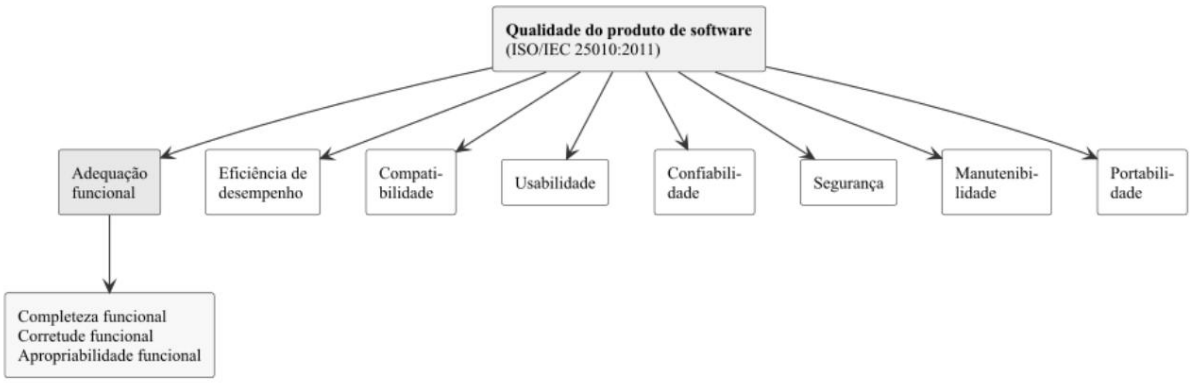
Figure 1 – ISO/IEC 25010:2011 software product quality model, highlighting

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS - Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

for functional suitability



Source: Adapted from International Organization for Standardization (2011).

The risk, however, lies not only in the absence of testing, but in conducting tests without a method. *Ad hoc* approaches , lacking a defined protocol, result in lower coverage and fewer defects. detected and additional development and maintenance costs — a particularly critical limitation. in governmental systems (Lima, 2023). Recent experiences in the Brazilian context confirm the gains obtained by replacing this informality with structured processes: Lima (2023) elaborated Functional test cases for the Rio Grande do Norte State Government's payment API, exceeding a previous approach without protocol; Gomes (2024) built an automation strategy for The Court of Auditors of the State of Rio Grande do Norte revealed real flaws, including... security; and Fernandes (2020) demonstrated the feasibility of Selenium WebDriver in the automation of Tests in institutional systems. These works, however, are concentrated in state agencies and federal. This creates the gap that motivates this study: the absence, in the national literature, of reports on the application of a documented and automated functional testing process — articulating Documentation standards, quality models, and automation — for municipal management systems. integrated into smart city initiatives, a context in which staff, budget and other constraints Process maturity levels differ substantially between the state and federal spheres.

Given this scenario, the question guiding this work is whether the functional requirements of Conecta are met. Canaã's services are provided as specified, in order to guarantee the reliability required by management. digital public. The overall objective is to subject the system to functionality tests that ensure the Meeting the operational needs of the Municipality of Canaã dos Carajás. Specifically, The intention is to plan and design the tests according to the IEEE 829-2008 standard; implement them with the Selenium *framework ;* run them in the three relevant user profiles and operational scenarios; Analyze the results in light of the functional adequacy of ISO/IEC 25010; and document the process accordingly. replicable protocol. Thus, the three pillars that support the approach are articulated: IEEE 829-2008, which provides the documentation and implementation protocol; ISO/IEC 25010, which defines the criteria.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

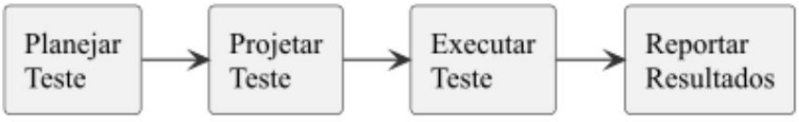
of quality; and Selenium, which enables automation. It should be acknowledged that IEEE 829-2008 was formally replaced by the ISO/IEC/IEEE 29119 family (International Organization for Standardization, 2022); its adoption is justified by the simplicity and conciseness of its artifacts. documentary — more suitable for the size of the project and the team involved than the structure multipartite nature of the successor standard—due to the broad base of literature and tools that reference it and because their documents are directly mappable to the corresponding artifacts of ISO/IEC/IEEE 29119-3, which preserves the methodological validity of the protocol.

2 MATERIALS AND METHODS

This research is characterized as a technical case study of an applied nature — an evaluation. empirical software analysis conducted on a real system, using a quantitative and descriptive approach, without manipulation of variables, comparison groups, or experimental hypotheses—and is structured around the four stages recommended by the IEEE 829-2008 standard — planning, design, execution and analysis of results—which constitute both the framework and the procedure of the study.

The standard provides a standardized set of guidelines for documenting and performing tests of software, promoting consistency, traceability and communication among those involved (IEEE) (Computer Society, 2008). Figure 2 summarizes the sequence of these steps: in the *planning,* The objective, approach, items under test, resources, and risks are defined; in the *project,* cases are specified . Testing, data, and environment; during *execution,* the cases are applied and the results are recorded, each being... Divergence is treated as a defect; and in the *analysis,* the data are evaluated to attest to the effectiveness of system. As an interpretative lens for this final stage, the functional adequacy of ISO/IEC was adopted. 25010, whose sub-characteristics — completeness, correctness, and appropriateness — guide the judgment on compliance with requirements (Estdale & Georgiadou, 2018).

Figure 2 – Partial flow of the IEEE 829-2008 process



Source: Adapted from IEEE Computer Society (2008).

2.1 Conecta Canaã System

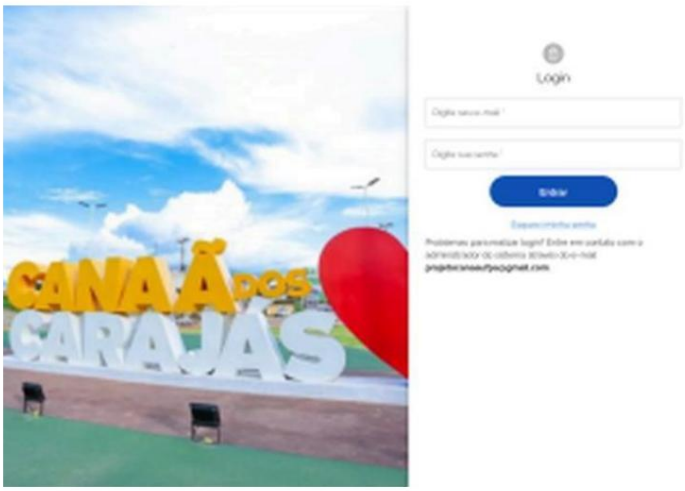
This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

Conecta Canaã Web1 is a public management environment that allows PMCC to administer
Requests for urban maintenance, reports of infractions, and demands related to the Code of Conduct.
of Canaã dos Carajás (Municipal Law No. 694/2015) and to the Ombudsman's Office. Its functions are distributed-
if divided into three user profiles: the **Executor,** responsible for resolving the incidents directed to him/her;
The **Controller,** who designates enforcers, interacts with citizens and changes the status of incidents;
and the **Administrator,** responsible for preparing reports and registering new users. Developed in
Using JavaScript and TypeScript with the React library in the interface layer, the system adopts an architecture
Three-tier client-server architecture — presentation (Google Chrome browser), logic (server)
Ubuntu with NGINX) and data (MongoDB 7.0) — and is hosted on the cloud infrastructure of
municipality. Figure 3 shows its access screen. The tests reported in this article were planned and
Executed in August 2024, on the then-current version of the system in a testing environment.
The system received updates after that date, so the results reported here are...
These figures should be read as a snapshot of the version evaluated in August 2024, and not of the current state of the...
platform.

Figure 3 – Login screen of the Conecta Canaã system



Source: Author (2024).

2.2 Test planning

The planning phase began with identifying the system's functional requirements, derived from...
user needs and expectations (FNR Machado, 2015; Sommerville, 2011). The
The survey identified thirty functional requirements across the system as a whole, distributed across the three

[1] Available at: https://conecta.smartcitycanaadoscarajas.com.br/

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

user profiles. Given this volume, the selection was based on three criteria — frequency of use,
relevance to the core functioning and complexity of the requirement—applied by the author himself,
based on their knowledge of the PMCC system and operation. *Frequency of use* considered
The recurrence of the functionality in the routine of the three profiles; *relevance* prioritized the requirements without the
which would render the central workflow for recording, processing, and monitoring incidents unfeasible;
and *complexity* favored functionalities with a greater number of interaction steps and rules of
Validation, due to the higher risk of defects. A numerical scale was not adopted, nor was assignment of
There were no weights or formal decision matrix, nor was there participation from other evaluators in the prioritization.
— limitation noted in Section 4.1. The process resulted in the seven requirements distributed across the three
The profiles presented in Table 1 represent approximately one-quarter of the total requirements.
System functionalities.

Table 1 – Selected functional requirements

Profile	ID	Functional Requirement
Controller	RF 01	Add Comment
Executor	RF 02	View Incident
Executor	RF 03	View Comment
Administrator	RF 04	Log in
Administrator	RF 05	Register User
Administrator	RF 06	Apply Search Filters
Administrator	RF 07	View Incidents

Source: Author (2024).

The definition of the test cases was based on the corresponding use cases, which describe the...
functionalities from the user's perspective and facilitate the derivation of tests capable of encompassing all
the requirements (Sommerville, 2011). Figures 4 and 5 illustrate, respectively, the use cases of
User registration and occurrence listing. As Craig and Jaskiel (2002) point out, a case of
The test specifies a condition to be verified, with input values, execution constraints, and...
expected result; in the adopted process model (Sommerville, 2011), summarized in Figure 6,
This specification constitutes the initial step in the implementation of the tests. From this mapping,
Three test cases were developed for the Executor and Controller profiles, and nine for the...
Administrator, totaling twelve, whose correspondence with the requirements is detailed in Table 2.

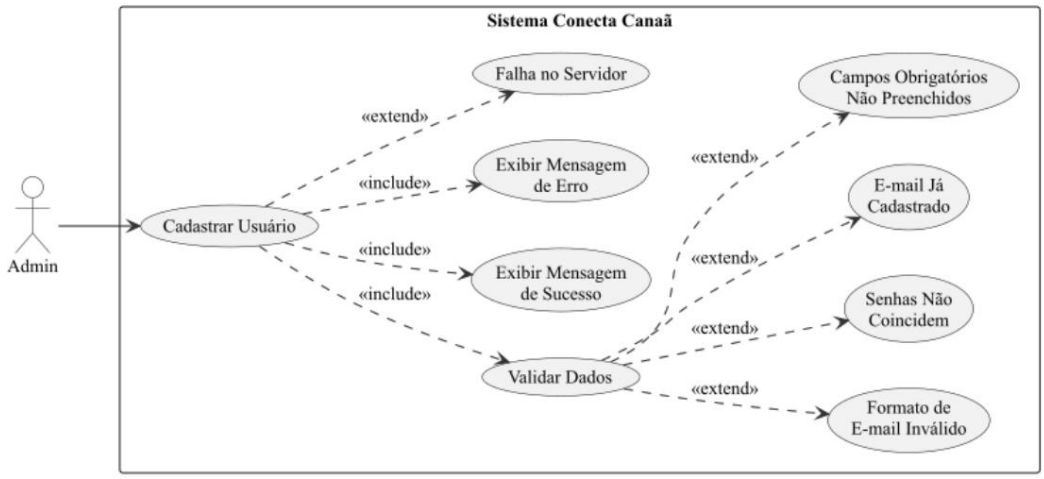
This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

In quantitative terms, the suite covers the seven selected requirements with an average of 1.7 cases per Requirement: RF 01 to RF 03 each have a main flow scenario; RF 04 to RF 06, with a There is one valid scenario and one invalid scenario each; and RF 07, with three visualization scenarios. Three of the twelve cases (25%) exercise exception flows — invalid credentials (TC 05), duplicate registration (TC 07) and search for non-existent code (TC 09) —, with all twelve cases having been executed in a way Automated. This version of the suite does not include systematic negative testing techniques such as Boundary value analysis, empty fields, malformed data, maximum lengths, characters special and invalid input combinations; this restriction resulted from the prioritization of flows. main and most frequent exceptions in the actual operation of the system, within the available timeframe for the study, and is acknowledged as a limitation in Section 4.1.

Figure 4 – User registration use case



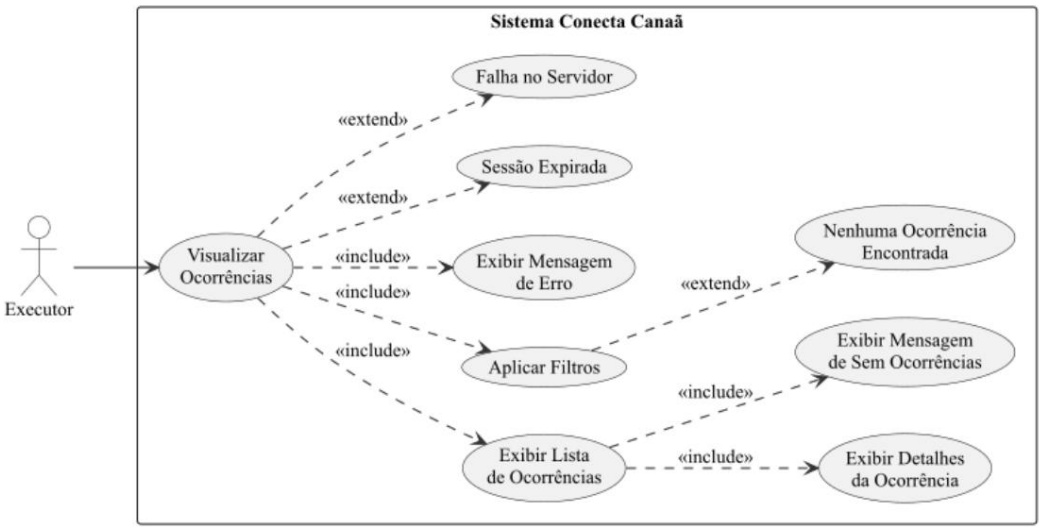
Source: Author (2024).

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS - Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

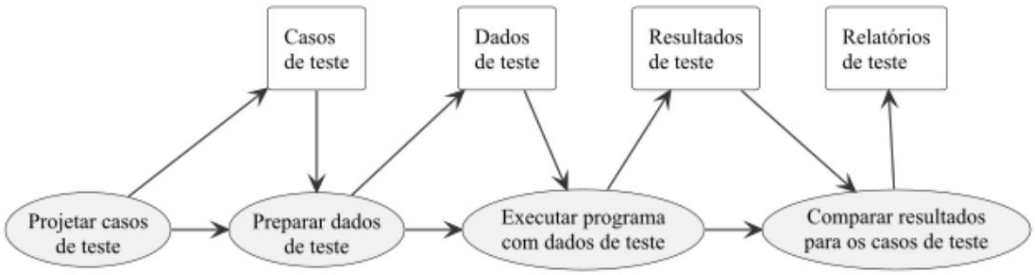
Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

Figure 5 – Use case for occurrence listing



Source: Author (2024).

Figure 6 – Software testing process model



Source: Sommerville (2011).

2.3 Test Design

Once the cases were defined, it was decided to implement them as automated functional tests. The activity Testing can be classified into four dimensions: the state of the test (unit, integration, or system), the technique employed (black-box, white-box, regression), the goal pursued (functional, of interface, performance, security) and the environment in which it is implemented, such as web applications. (Molinari, 2008). The present study is positioned within the functional goal, using the black-box technique, in web environment. Functional testing evaluates the software based on its specifications, without access to the web. source code — which is why it is also called black-box testing — looking for discrepancies. between observed and expected behavior (Myers et al., 2012). Its automation, which delegates the *scripts* handle execution and result verification, reducing human effort and speeding up repetition. It reduces costs (Bernardo & Kon, 2008; Dustin et al., 2009), a significant benefit when considering...

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

that testing activity can consume 30% to 35% of the total effort of a project (Mustofa & Fajar, 2018).

Table 2 – Mapping of test cases

RF	TC	Test Case
RF 01	TC 01	Add comment to the incident.
RF 02	TC 02	View citizen's incident report
RF 03	TC 03	View incident report comment
RF 04	TC 04	Login with valid credentials.
RF 04	TC 05	Login with invalid credentials
RF 05	TC 06	Register a new administrator user.
RF 05	TC 07	Register an existing user.
RF 06	TC 08	Apply code search filter
RF 06	TC 09	Search with no code available.
RF 07	TC 10	View open cases
RF 07	TC 11	View completed incidents
RF 07	TC 12	View incidents without location.

Source: Author (2024).

The automation was conducted using the Selenium *framework ,* a well-established reference for web applications. Conceived in 2004 as an alternative to the inefficiency of manual testing (Gomes, 2024), Selenium WebDriver accesses the page's DOM and interacts directly with its elements — clicks, Filling out forms and validating responses—faithfully replicating real-life actions. of the user (Fernandes, 2020; Romanini & Sotto, 2019). Its adoption was based on three factors: a Efficiency in covering multiple profiles and scenarios, compatibility with Google Chrome — system access platform — and integration with the Python language, used in script development .

The automation setup took into account the attributes that ISTQB associates with its effectiveness: the architecture. Test automation, aligned with the architecture of the system under test; minimizing dependencies. between the graphical interface and the interaction logic; and the choice of an intuitive and well-defined *framework .* documented, balancing costs, benefits and risks (ISTQB, 2016). The combination of Selenium and

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

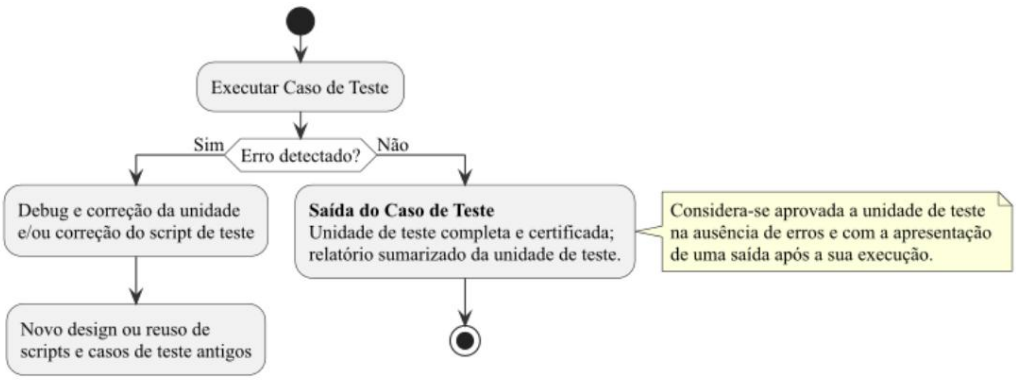
Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

Python meets these requirements, in addition to facilitating the maintenance and re-execution of *scripts* each time.
new version of the system.

The tests took place in a homologation environment isolated from production, a condition that prevented any...
interference with the municipality's actual data during verification. In this environment, the following were created:
Three user profiles were created, and a sample of 30 occurrences was extracted from the database, consisting of
20 open cases and 10 completed — the records without location information, required by TC 12,
They constitute a subset of these two groups, and not a third category, since the absence of
Geographic coordinates are an attribute independent of the stage of processing of the case. The composition
This thus covers the scenarios required by the *dashboard* test cases (TC 10 to TC 12).
The sample served as a reference to compare the values displayed by the interface with the records.
effectively stored, making the verification of functional appropriability objective. A
The description of the scenarios followed the standard form of Lino (2021), recognized for its objectivity.
The workflow adopted is shown in Figure 7: once a conclusive result is obtained, the case is closed.
and revised only in response to changes in functionality; otherwise, a report is generated.
The defect has been forwarded to the development team.

Figure 7 – Execution flow of test cases



Source: Lino (2021).

2.4 Execution environment and reproducibility

To enable independent replication of the tests, the conditions under which they were performed are recorded.
The scripts were developed in Python 3.10 with Selenium 4.21.0 and executed with the *framework.*
pytest 8.3.2 on Ubuntu 22.04 operating system, in the Google Chrome browser version
stable current at the time of execution (August 2024), with ChromeDriver provisioned.
automatically by the chromedriver-autoinstaller 0.6.4 package, which resolves the issue at runtime.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

The *driver* version is compatible with the installed browser — which is why the suite remains
Runs in later versions of Chrome without manual adjustment. Dependencies are declared.
in the requirements.txt file of the repository (Pessoa, 2024), and execution is done using the pytest command,
with environment parameters (system URL and test profile credentials) provided by
Environment variables. The correspondence between *scripts* and cases is documented in the file.
The repository's README file has an automated function for each of the twelve cases:
test_controller_report_executor.py covers TC 01 to TC 03; test_login.py, TC 04 and TC 05;
test_register_user.py, TC 06 and TC 07; test_occurrence_no_location.py, TC 08, TC 09 and TC 12; and
test_dashboard.py, TC 10 and TC 11. Test data — user profiles and sample occurrences
— were created synthetically in the homologation environment, with support from the Faker library for the
Generation of registration data without using real personal data. Among the executions, the database of
The homologation process was restored from a previously saved *dump ,* ensuring the initial state.
identical in each round and allowed the re-execution of cases that alter data, such as the registration of
user (TC 06). Each case was run five times, with consistent results across all cases.
Repetitions and registration in the standard form and generation of HTML reports by pytest-html-reporter.
versioned in the repository's report directory; the fivefold repetition of each case served, in itself,
as a control for unstable tests *(flaky tests),* no variation in outcome was observed between
the rounds.

2.5 Execution

The implementation took the form of automation *scripts* developed in Python with Selenium and versioned in a
public repository2 (Pessoa, 2024). Figure 8 illustrates the authentication routine:
The login function opens the provided URL, locates and fills in the email and password fields, and submits the...
The form checks for an element indicating successful access, returning True.
or False. Each case was documented according to the standard form, with code, profile, requirement.
associated, precondition, procedure, and expected outcome. Tables 3, 4, and 5 reproduce a
a representative example of each profile — the inclusion of a technical opinion from the Controller,
Restricted viewing of events by the Executor and the attempt to authenticate with credentials.
invalidated by the Administrator—while the remaining cases follow the structure mapped in Table 2.

[2] Available at: https://github.com/ramon141/-canaa-conecta-web-tester

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

Figure 8 – Automated authentication routine with Selenium

```
def login(browser, email, password) -> bool:
    browser.driver.get(os.getenv('HOME_PAGE'))

    email_field = browser.driver.find_element(
        By.XPATH, "//div[contains(@class,"
        "'MuiFormControl-root')][1]//input")
    email_field.clear()
    email_field.send_keys(email)

    password_field = browser.driver.find_element(
        By.XPATH, "//div[contains(@class,"
        "'MuiFormControl-root')][2]//input")
    password_field.clear()
    password_field.send_keys(password)

    browser.driver.find_element(
        By.XPATH, "//button[@type='submit']").click()

    try:
        browser.wait.until(EC.presence_of_element_located
        (
            (By.XPATH, "//*[contains(text(),"
             "'Ocorrencias Comuns')]")))
        return True
    except:
        return False
```

Source: Author (2024).

Table 3 – TC 01 – Add comment to the occurrence

Field	Description
Code	TC 01
Profile	Controller
RF	RF 01
Precondition	Being on the login screen
Procedure	1. Log in with controller credentials 2. Select the incident and click on View Details. 3. Fill in the technical opinion field. 4. Click on Perform Opinion 5. Confirm shipment
Expected result	"Report successfully registered."
Date	16/08/2024
Result	Approved

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

Source: Author (2024).

Table 4 – TC 02 – View citizen occurrence

Field	Description
Code	TC 02
Profile	Executor
RF	RF 02
Precondition	Being on the login screen
Procedure	1. Log in using executor credentials.
Expected result	View only the occurrences directed to the specific executor.
Date	16/08/2024
Result	Approved

Source: Author (2024).

Table 5 – TC 05 – Login with invalid credentials

Field	Description
Code	TC 05
Profile	Administrator
RF	RF 04
Precondition	Being on the login page
Procedure	1. Invalid email address entered. 2. Invalid password entered. 3. Click the Enter button.
Expected result	"There was a problem logging in, please check your credentials!"
Date	16/08/2024
Result	Approved

Source: Author (2024).

3 RESULTS

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS - Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

The execution of the twelve test cases followed the planned procedure completely, and their outcomes...
The data were consolidated in Table 6, organized by functional requirement and user profile. All of them
The cases were approved, resulting in a 100% success rate.
In the Controller profile, TC 01 confirmed the inclusion of a technical opinion in an incident: after the
After filling in the technical field and confirming the submission, the system returned the message “Opinion”.
"Registered successfully.", fulfilling RF 01. In the Executor profile, TC 02 and TC 03 showed...
that each user exclusively views the occurrences and comments directed to them, without
Access to the records of other executors, which confirms the preservation of isolation between profiles.
as foreseen in RF 02 and RF 03.
The highest number of cases were concentrated in the Administrator profile, distributed across four requirements.
The authentication module accepted the valid credentials, leading to the *dashboard* (TC 04), and
The invalid ones were blocked with the corresponding alert message (TC 05), in accordance with RF 04. The
The user registration process functioned correctly when registering a new administrator (TC 06) and when rejecting the
existing email record (TC 07), preserving the uniqueness required by RF 05. The filters of
The search, in turn, returned accurate results when a valid code (TC 08) was consulted and
They indicated a lack of correspondence when the code did not exist (TC 09), in accordance with
RF 06.

Finally, the cases associated with RF 07 verified the *dashboard's accuracy:* the filter was applied.
During the period between 01/01/2021 and 16/08/2024, the system recorded 20 open occurrences (TC 10) and
10 completed (TC 11) — totaling the 30 in the sample. —, in addition to correctly displaying the occurrences
without recorded location (TC 12), a subset of the two previous groups. In all cases, the
The values displayed by the interface matched the 30 records extracted from the database.
Approval, without discrepancies.

Table 6 – Consolidated results of the test cases

TC	RF	Description	Res.
TC 01	RF 01	Add comment	OK
TC 02	RF 02	View occurrence	OK
TC 03	RF 03	View comment	OK
TC 04	RF 04	Valid login	OK
TC 05	RF 04	Invalid login	OK
TC 06	RF 05	Register user	OK

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

TC	RF	Description	Res.
TC 07	RF 05	Duplicate user	OK
TC 08	RF 06	Filter by code	OK
TC 09	RF 06	Filter does not exist.	OK
TC 10	RF 07	Open incidents	OK
TC 11	RF 07	Completed incidents	OK
TC 12	RF 07	No location	OK

Source: Author (2024).

4. DISCUSSION

The full approval of the test cases should be interpreted precisely: it means that all twelve
The cases passed the evaluated conditions, and it is within this scope that the interpretation by sub-characteristics occurs.
Regarding the functional adequacy of ISO/IEC 25010 (Estdale & Georgiadou, 2018). The most robust evidence
This refers to *functional correctness* —the accuracy of the results produced—supported by both tests.
authentication (TC 04 and TC 05) and registration with uniqueness verification (TC 06 and TC 07), which
They confirmed the correct behavior in both valid and exception flows, as well as by coincidence.
between the values displayed by the *dashboard* and the records in the homologation database (TC 10 to TC 12).
Regarding *functional completeness* — the scope of functions in relation to the user's tasks — the
The results offer only partial evidence: the twelve cases cover the seven prioritized requirements.
but not the entire universe of system functionalities, so completeness cannot be claimed.
Functional suitability of the product as a whole. Similarly, *functional suitability* — suitability
from the functions to the specified objectives — finds favorable evidence in the fact that the evaluated flows
to fulfill the tasks for which they are intended, but full demonstration of this would require evaluation by the
end users, which goes beyond the scope of this study.

These findings confirm the effectiveness of a structured four-step testing process in the face of
informality of *ad hoc approaches,* pointed out by Lima (2023) as a source of low coverage and
reliability in government systems. Automation with Selenium has extended this gain to
covering multiple profiles with a smaller margin of human error and greater accuracy than verification
manual (Romanini & Sotto, 2019), in accordance with the ISTQB guidelines on the relationship between
cost, benefit and risk (ISTQB, 2016).

A comparison with the literature situates the contribution along two axes. Regarding technical feasibility, the

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

The results are consistent with those of Fernandes (2020), who confirmed the suitability of Selenium.
WebDriver has been used to automate testing in institutional web systems, and this evidence is now being extended.
to the domain of municipal management. As for the outcome, the contrast with Gomes (2024) is enlightening:
While the automation applied to the Court of Auditors of Rio Grande do Norte revealed real flaws,
Including security, Conecta Canaã did not report any functional failures. This difference does not diminish the...
The value of the procedure — conversely, a set of tests that finds no defects is only
informative when derived from a rigorous protocol, capable of distinguishing the absence of failures from
absence of verification — and finds a plausible explanation in the maturity state of the system.
Evaluated: At the time of testing, Conecta Canaã had been in production for about two years.
with more than 2,600 processed occurrences (Bacelar et al., 2025), so that the main flows
The methods used here had already been extensively tested—and refined—through real-world use. Some-
This is what the suite prioritized, focusing on frequent main and exception flows, scenarios in which defects...
Remnants are less likely in a stable system, which reinforces Andrea's argument.
(2007) on the role of systematic testing in product quality without authorizing generalizations
Regarding the absence of defects.

Beyond defect detection, the adoption of IEEE 829-2008 added value through standardization.
Documentary. The formalization of each case into code, profile, associated requirement, precondition,
The procedure and expected result established explicit traceability between requirements and...
Test evidence, a condition that supports audits, facilitates communication within the team and allows
Re-execute the verification with each system change — a benefit unavailable with informal approaches.
This traceability is particularly relevant in public administration, where accountability and...
Transparency is an integral part of the system's purpose.
In terms of implications, the result documents a case of application of a structured process of
Tests in the municipal public management system contribute to densification—which is still incipient—
of Brazilian smart city initiatives (Lazzaretti et al., 2019); by ensuring verification
By systematically addressing critical functionalities, the process can indirectly contribute to trust.
public in digital services, a critical element of their legitimacy (Purbaratri & Krisnawaty, 2023).
although this relationship was not measured in this study.

4.1 Threats to validity

Regarding *construct validity,* the 100% approval rate only measures the conformity of the twelve.
cases under the evaluated conditions; it does not constitute a measure of robustness, overall reliability, or
absence of defects, and the association with the sub-characteristics of ISO/IEC 25010 is restricted to

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS - Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

as stated at the beginning of this section. Regarding *internal validity,* the tests were conducted in an environment of approval, based on a controlled sample of 30 occurrences and synthetic data, conditions not necessarily equivalent to the production environment under real load; furthermore, the *scripts* were Created and executed by the author himself, without independent verification. The selection of seven The requirements were also determined individually, without numerical scaling, explicit weights, or consensus. among reviewers, which makes it susceptible to judgment bias and limits the demonstration of representativeness in relation to the system's overall functionalities. Regarding *external validity,* it deals with... If it is a single case study, about a specific system and municipality: the protocol is replicable, However, the results are not generalizable to other systems without further application. Regarding the *validity of In conclusion,* although each case was repeated five times with stable results—which reduces the risk of random outcomes or unstable tests —, the small number of cases (twelve) and the absence The experimental variation between repetitions prevents any inferential statistical treatment of the... Results. Finally, the limitations of the adopted automation are noted: the Selenium *scripts* depend of the interface selectors — *layout* changes require maintenance —, check the behavior observable by the browser without inspecting the server logic, and the suite does not incorporate techniques Negative testing systems, such as boundary value analysis and malformed data, as per discussed in Section 2.2.

5 CONCLUSIONS

This work systematically applied functionality tests to the Conecta Canaã web system. articulating the IEEE 829-2008 standard as a protocol for planning, design, execution, and analysis; ISO/IEC 25010 as a quality reference; and the Selenium *framework* as a tool for automation. The approval of the twelve test cases across the three user profiles demonstrates that the system Under the evaluated conditions, it meets the seven selected functional requirements, operating in accordance with the... Specifications within the verified scope.

Three contributions result from the study: the empirical demonstration that IEEE 829-2008 is applicable. and effective for documenting and conducting tests on municipal e-government systems; the production evidence of functional correctness of Conecta Canaã — with partial indications of completeness and appropriability, restricted to the evaluated requirements — according to ISO/IEC 25010; and availability of a replicable protocol for automated testing of similar-sized systems, aligned with agenda for modernizing public management and the Sustainable Development Goals (Souza) et al., 2025). More than just technical improvement, the structured process ensured traceability between Requirements and test evidence are attributes that support audits and can contribute to trust.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

of citizens in digital public services (Purbaratri & Krisnawaty, 2023).

However, limitations are acknowledged: the tests were restricted to the homologation environment and to a...
controlled sample; the twelve cases cover seven functional requirements, without exhausting all functionalities.
of the system; and the scope was limited to the functional dimension, without considering performance, security and
usability. As a further step, the incorporation of systematic testing techniques is suggested.
Negative — boundary value analysis, empty fields, malformed data, and invalid combinations.
—, the expansion to the non-functional requirements of the other characteristics of ISO/IEC 25010, the
integrating tests into a continuous integration and continuous delivery *pipeline ,* as exemplified by the experience of
Gomes (2024), and the replication of the methodology in other systems of the smart cities network of
Pará, enabling comparative quality studies between municipalities.

REFERENCES

Andrea, J. (2007). Envisioning the Next-Generation of Functional Testing Tools. *IEEE Software, 24.* https://doi.org/10.1109/MS.2007.73

Bacelar, P., Broechl, G., Sousa, M., Kuribayashi, H., Vijaykumar, N., & Francês, CR (2025). Integrated System Connects Canaã: An Architectural Solution for Mitigating the Difficulties of Emerging Urbanization. *Proceedings of the XXVII Ibero-American Congress on Software Engineering (CIbSE).*

Bartié, A. (2002). *Software quality assurance: acquiring organizational maturity* (1st ed.). Elsevier.

Batty, M., et al. (2012). Smart cities of the future. *The European Physical Journal Special Topics,* 214(1), 481–518. https://doi.org/10.1140/epjst/e2012-01703-3

Bernardo, PC, & Kon, F. (2008). The Importance of Automated Testing. *Software Engineering Magazine,* 1(3), 54–57. Retrieved August 25, 2024, from https://www.ime.usp.br/~kon/papers/EngSoftMagazine-IntroducaoTestes.pdf

Craig, R. D., & Jaskiel, S. P. (2002). *Systematic Software Testing* (1st ed.). Artech House Publishers.

Dustin, E., Garrett, T., & Gauf, B. (2009). *Implementing automated software testing: how to save time and lower costs while raising quality* (1st ed.). Addison-Wesley Professional.

Estdale, J., & Georgiadou, E. (2018). Applying the ISO/IEC 25010 Quality Models to Software Product. *Systems, Software and Services Process Improvement. EuroSPI 2018, 896,* 492–503. https://doi.org/10.1007/978-3-319-97925-0_42

Fernandes, GS (2020). *Test automation with Selenium WebDriver applied to the course distribution system* [Undergraduate Thesis (Bachelor of Information Systems)]. Federal University of Uberlândia, Faculty of Computing. Retrieved July 13, 2026, from https://repositorio.ufu.br/handle/123456789/31143

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

Gomes, RL (2024). *Development and Evaluation of a Cypress-Based Test Automation Strategy for the TCE-RN* [Final Project (Specialization in Applied Information Technology)]. Federal University of Rio Grande do Norte, Digital Metropolis Institute.

IEEE Computer Society. (2008). *829-2008 — IEEE Standard for Software and System Testing.* IEEE Computer Society. Fredericksburg, VA, USA.

Brazilian Institute of Geography and Statistics. (2022). 2022 Census — Canaã dos Carajás. Retrieved January 1, 2025, from https://www.ibge.gov.br/cidades-e-estados/pa/canaa-dos-carajas.html

International Organization for Standardization. (2011). *ISO/ IEC 25010:2011 — Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.* ISO/IEC. Geneva.

International Organization for Standardization. (2022). *ISO/ IEC/ IEEE 29119-1:2022 — Software and systems engineering — Software testing — Part 1: General concepts.* ISO/IEC/IEEE. Geneva.

ISTQB. (2016). *Advanced Level Syllabus Test Automation Engineer Version 2016.* International Software Testing Qualifications Board.

Lazzaretti, K., Sehnem, S., Bencke, F.F., & Machado, H.P.V. (2019). Smart cities: insights and contributions from Brazilian research. *Urbe. Brazilian Journal of Urban Management, 11,* e20190118. https://doi.org/10.1590/2175-3369.011.001.e20190118

Lima, KK d. (2023). *Development and Implementation of Functional Test Cases for the PagRN API* [Final Project (Specialization in Applied Information Technology)].
Federal University of Rio Grande do Norte, Digital Metropolis Institute.

Lino, ADP (2021). *LabDER — Virtual Laboratory for Teaching and Learning Relational Databases: An approach to automatic evaluation of ER Diagrams and SQL* [Doctoral Thesis].
University of Coimbra. August 2023. https://hdl.handle.net/10316/95440 Obtained 25, 2024,

Machado, FNR (2015). *Software Requirements Analysis and Management — Where systems are born.* (3rd ed.). Érica Publishing House.

Machado, R. (2023). Smart cities and public management: an exploratory relationship of the Brazilian scenario. *Brazilian Journal of Public Management,* 2(2), 1–9.

Molinari, L. (2008). *Software Testing — Producing better and more reliable systems* (4th ed.).
Erica Publishing.

Mustofa, K., & Fajar, S. P. (2018). Selenium-Based Multithreading Functional Testing. *Indonesian Journal of Computing and Cybernetics Systems,* 12(1), 63–72. https://doi.org/10.22146/ijccs.28121

Myers, G. J., Sandler, C., & Badgett, T. (2012). *The art of software testing* (3rd ed.). John Wiley & Sons.

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability Maturity Model for Software, Version 1.1. *IEEE Software,* 10(4), 18–27. https://doi.org/10.1109/52.219617

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

Year VII, v.2 2026 | Submission: October 10, 2026 | Accepted: October 15, 2026 | Publication: October 29, 2026

Pessoa, RB (2024). Canaã Conecta Web Tester. Retrieved November 19, 2024, from https://github.com/ramon141/-canaa-conecta-web-tester

Pressman, RS, & Maxim, BR (2021). *Software Engineering: A Practitioner's Approach* (9th ed.). AMGH.

Purbaratri, W., & Krisnawaty, N. (2023). Software Quality Model ISO/IEC 25010 Prioritization Smart Government Application. *Proceeding of the Perbanas International Seminar on Economics, Business, Management, Accounting and IT (PROFICIENT),* 261–270.

Rios, E., & Moreira, T. (2006). *Software Testing* (2nd ed.). Alta Books Editora.

Romanini, IZ, & Sotto, ECS (2019). Selenium WebDriver in the Evolution of Manual Testing. *Technological Interface Magazine,* 16(2), 313–322. https://doi.org/10.31510/infa.v16i2.627

SOFTEX. (2016). *MPS General Software Guide.* Association for the Promotion of Excellence in Brazilian Software.

Sommerville, I. (2011). *Software Engineering* (9th ed.). Pearson Prentice Hall.

Souza, MJC d., Galhardo, CX, Teixeira, K., & Moreira, A. (2025). Development and Validation of Software for Automation of Technological Management Processes in the Public Service. *Revista Multidisciplinar do Nordeste Mineiro,* 21(3), 1–29. https://doi.org/10.61164/s4608v39

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.