Privacy Risks Associated with the Use of LLMs in Software Development

Diego Menegazzi[1], **Edna Dias Canedo**[1]

[1]University of Brasília (UnB), Department of Computer Science
Brasília, DF, Brazil

omenegazzi@gmail.com, ednacanedo@unb.br

Abstract. ***Research Context****: Large Language Models (LLMs), e.g., GPT-2 and GPT-4, are increasingly embedded in software development to assist with code generation, testing, and documentation. Adoption promises productivity gains but also raises challenges for Information Systems (IS), particularly data privacy and regulatory compliance.* ***Scientific and/or Practical Problem****: Developers may inadvertently include personal or sensitive information in prompts. LLMs can temporarily retain and reproduce this data in later outputs, causing privacy breaches. This creates technical, ethical, and legal concerns under the Brazilian LGPD and the European GDPR. While recent studies survey developer perceptions, empirical evidence showing violations in practice remains scarce.* ***Proposed Solution and/or Analysis****: We present an experiment simulating a Membership Inference Attack (MIA) on a local GPT-2 instance. Using synthetic personal data with realistic identifiers, we evaluate how sensitive information can be recalled by the model and propose mitigation strategies aligned with regulatory and organizational frameworks.* ***Related IS Theory****: The study is grounded in the Socio-Technical Systems Theory and Information Systems models such as the People-Process-Technology (PPT) framework, emphasizing the alignment of human, organizational, and technological dimensions. It also connects to the GranDSI-BR 2016–2026 research challenges on privacy, ethics, and security in intelligent IS.* ***Research Method****: We adopted a mixed-methods approach, combining an experimental MIA simulation on GPT-2 with a literature review and regulatory analysis. Synthetic Brazilian personal data ensured realistic and ethically compliant experimentation.* ***Summary of Results****: The experiment showed that GPT-2 can reproduce sensitive identifiers from prior prompts, even without fine-tuning or persistent memory.* ***Contributions and Impact to IS area****: The study provides reproducible experimental evidence of privacy risks in LLMs and reinforces the urgency of embedding privacy-by-design principles into IS development workflows.*

1. Introduction

Large Language Models (LLMs), such as GPT-2, GPT-4, GPT-5 and LLaMA, are transforming Information Systems (IS) by enabling intelligent automation of tasks that traditionally required extensive human effort. In software development workflows, these models are increasingly adopted to support code generation, bug fixing, documentation, and testing [Liu et al. 2024, Yao and Zhang 2023]. Their integration into digital ecosystems is not restricted to the technical dimension: organizations view LLMs as strategic tools to optimize processes, reduce costs, and expand innovation capacity across multiple domains, including government, education, and healthcare.

However, alongside these benefits, critical risks arise when developers insert sensitive or personal information into prompts. Unlike conventional tools, LLMs operate with probabilistic reasoning over context windows, which may temporarily retain identifiers and reproduce them in subsequent outputs. This introduces scenarios where confidential data, such as tax identifiers, medical information, or corporate secrets, can be inadvertently exposed, violating fundamental principles of privacy and security. A widely reported case was Samsung Electronics, where employees leaked confidential source code through ChatGPT queries [Mauran 2023]. Such incidents highlight the practical consequences of underestimating privacy risks in real development environments.

From a socio-technical perspective, the problem is amplified by organizational practices. Many teams adopt LLMs informally, without clear governance or guidelines, often reusing prompts, logging conversations, or sharing model instances across departments. These practices conflict with regulatory obligations such as Brazil's General Data Protection Law (LGPD) [LGPD 2018] and the European GDPR [GDPR 2016], which mandate data minimization, purpose limitation, and accountability. Inadequate controls may not only result in data breaches but also compromise organizational trust, legal compliance, and the broader societal expectation of responsible use of AI-based IS.

Falcão and Canedo [Falcão and Canedo 2024] investigated Brazilian software professionals' perceptions of data privacy in the use of LLMs. Through a survey with 78 participants from all regions of Brazil, we analyzed their knowledge of the LGPD, perceived risks, mitigation practices, and organizational challenges. The results revealed that most developers possessed only basic understanding of privacy issues, rarely employed anonymization techniques, and considered current legislation insufficient to address the risks posed by LLMs. As contributions, that study proposed practical recommendations, emphasizing training, organizational guidelines, and the need for supporting tools to ensure ethical and secure LLM adoption. While the survey provided valuable insights into perceptions, it lacked experimental evidence of how privacy risks materialize in practice.

The lack of experimental studies addressing privacy risks in LLMs constitutes the central research problem of this work. Specifically, it remains unclear how and under what conditions LLMs reproduce sensitive information introduced during development tasks, even in local or offline settings. This gap hinders the ability of IS researchers and practitioners to design effective safeguards and undermines the socio-technical alignment of people, processes, and technologies advocated in IS theory.

To address this gap, this paper presents an experiment simulating a Membership Inference Attack (MIA) against a local GPT-2 model. Using synthetic data with realistic Brazilian identifiers, we demonstrate how sensitive information provided in prompts can be recovered by subsequent queries, despite the absence of fine-tuning or persistent memory. Figure 1 illustrates this scenario: a developer enters personal data (e.g., CPF) into an LLM prompt, and the model later reproduces the data in response to a probing query, thereby simulating a privacy leak.

The contributions of this paper are threefold: (i) it provides empirical evidence of privacy vulnerabilities in LLMs applied to software development, complementing

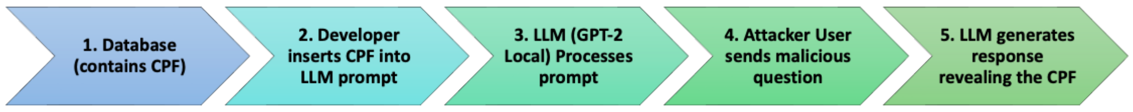

1. Database (contains CPF)
2. Developer inserts CPF into LLM prompt
3. LLM (GPT-2 Local) Processes prompt
4. Attacker User sends malicious question
5. LLM generates response revealing the CPF

Figure 1. Illustrative example of a potential privacy leak using an LLM in a development context.

perception-based research with reproducible experimentation; (ii) it aligns the findings with the GranDSI-BR 2016–2026 research challenges, particularly in the areas of privacy, security, and ethics in intelligent IS; and (iii) it advances IS theory and practice by emphasizing privacy-by-design as a necessary principle when adopting LLMs in socio-technical contexts that involve people, processes, and technologies.

The remainder of this paper is structured as follows: Section 2 presents the theoretical background and related work on LLMs, privacy risks, and IS regulatory frameworks. Section 3 details the research design and experimental setup. Section 4 reports and analyzes the experimental findings. Section 5 discusses implications for IS, compares our results with related work, and outlines ethical and organizational considerations. Section 6 presents threats to validity and limitations. Finally, Section 7 concludes with final remarks and directions for future research.

2. Background and Related Work

2.1. LLMs in Software Development

Large Language Models (LLMs) have gained widespread popularity for their ability to perform tasks such as code generation, bug detection, and documentation generation. These capabilities make them attractive tools for enhancing software development productivity [Liu et al. 2024, Nam and Kim 2024]. LLMs typically use transformer-based architectures and are trained on large volumes of data, including source code, technical documentation, and various types of web content [Yao and Zhang 2023].

Large Language Models have been incorporated into software workflows to support code generation, testing, and documentation. Studies such as Kim and Hua (2025) and Birru et al. (2025) demonstrate productivity gains, but also highlight the need for human oversight and governance when LLMs interact with real organizational data. These works focus primarily on output quality and reliability rather than on privacy leakage during developer prompts, which is the central concern of this study.

Several studies have explored how LLMs are integrated into development workflows. Falcão and Canedo [Falcão and Canedo 2024] investigated how software team members perceive privacy risks when using LLM-based tools. Their study found that while developers recognize the usefulness of these tools, many are unaware of the privacy implications of inputting sensitive data into LLMs. This gap between perceived usefulness and awareness of privacy risks highlights a significant vulnerability in the use of these tools.

Beyond perceptions of risk, recent research has also focused on systematically evaluating the quality of LLM outputs across software engineering tasks. Kim and Ming [Kim and Hua 2025] conducted a comparative case study of five prominent LLMs (Chat-

GPT, Claude, Copilot, Gemini, and Meta) to assess the *reliability*, defined as the validity and consistency of generated software artifacts, and the *similarity* of their outputs. Their findings showed that while models such as Claude and ChatGPT achieved high reliability scores (0.92 and 0.90, respectively), overall similarity across models remained moderate at 57%, suggesting that each system exhibits distinct strengths and weaknesses. Importantly, the study emphasized that although LLMs can effectively support tasks ranging from requirements analysis to testing, they still require substantial human oversight, particularly in later phases such as implementation and testing where reliability declines.

Complementing these findings, Birru et al. [Birru et al. 2025] proposed CodeDocSync, an approach that leverages LLMs to automate technical documentation updates in Continuous Software Development (CSD) contexts. Applied in an industrial case study at Nokia, CodeDocSync integrates Retrieval-Augmented Generation (RAG), prompt engineering techniques (e.g., emotion prompting), and a chat interface to assist technical writers. The system achieved high answer relevancy (0.86 with GPT-3.5-turbo, up to 0.94 with emotion prompting) and strong factual consistency, demonstrating the potential of LLMs to reduce manual effort and maintain documentation quality in fast-paced release cycles.

Martin et al. [Martin et al. 2025] extended this discussion to the domain of requirements engineering, evaluating the potential of LLMs in classifying security-related software requirements. By comparing LLMs against traditional machine learning techniques on datasets such as SecReq, DOSSPRE, and PROMISE+, they demonstrated that LLMs can achieve competitive accuracy without domain-specific fine-tuning. Their results highlight that LLMs not only support code and documentation tasks but also contribute to critical activities in requirements engineering, particularly in the identification of non-functional requirements related to security.

Together, these studies demonstrate both the promise and the limitations of LLMs in software development: they can improve productivity and broaden support across the lifecycle, but they introduce new challenges regarding privacy, consistency, documentation quality, and requirements classification. However, most of these works focus on output quality, reliability, or developer perceptions, rather than on the concrete risk of personal data exposure when developers include PII directly in prompts during routine activities. This gap, privacy leakage at prompt time in everyday developer interactions, is the central concern addressed in this study.

Although the studies above demonstrate productivity and reliability gains, none of them experimentally evaluate privacy leakage during prompt-time developer interactions. This gap motivates the present study.

2.2. Privacy Risks and Attack Vectors in LLMs

The literature on LLM privacy can be organized into three main strands: (i) training-time attacks, such as classical Membership Inference and model inversion aimed at discovering whether specific data were part of the training corpus [Shokri et al. 2017, Yan et al. 2024]; (ii) prompt-based attacks, including prompt injection and jailbreaking, which manipulate model behavior to bypass safeguards or elicit restricted content [Liu et al. 2024, Choquet et al. 2024]; and (iii) perception and governance studies that analyze how practitioners and organizations understand and manage LLM risks

[Falcão and Canedo 2024, Gonçalves et al. 2024].

While these works provide valuable taxonomies and defenses, most focus either on adversarial scenarios or on training data leakage. The situation in which a developer unintentionally exposes PII during routine interactions with an LLM remains underexplored, particularly in local or offline deployments such as the one evaluated in this paper.

2.3. Legal and Socio-Technical Perspective

Under the Brazilian LGPD and the European GDPR, principles such as data minimization, purpose limitation, and security impose strict constraints on how personal data can be processed. Rocha et al. [Rocha et al. 2023] and Ferrão et al. [Ferrão et al. 2024] show that Brazilian software teams face difficulties translating these principles into development practices. De Cerqueira et al. [de Cerqueira et al. 2022] propose the RE4AI guide to operationalize ethical requirements, emphasizing the need for actionable support in agile contexts.

These studies reveal a socio-technical gap: organizations recognize the importance of privacy, yet developers often lack concrete mechanisms to prevent exposure when using AI assistants. This gap is precisely where our study is positioned.

2.4. How This Work Differs

This research differs from previous studies in three fundamental aspects:

- Moment of leakage: Prior MIAs typically target training datasets or large-scale cloud services. We evaluate *prompt-time leakage* occurring during ordinary developer interactions, even in a local GPT-2 instance without fine-tuning or persistent memory.
- Context of use: Instead of adversarial exploitation, we reproduce benign developer behavior (testing and report generation with structured PII), reflecting realistic IS workflows where privacy breaches may occur without malicious intent.
- Evaluation strategy: We employ a controlled experiment with synthetic Brazilian identifiers and an explicit protocol (Exact/Partial/No Leak), enabling objective linkage between technical behavior and LGPD/GDPR principles.

Therefore, our contribution is not another taxonomy of attacks, but reproducible empirical evidence that everyday use of LLMs can violate privacy-by-design principles in Information Systems.

2.5. Positioning within IS Research

Table 1 summarizes representative studies and highlights the shift proposed in this work from training-centric analyses to developer-centric prompt interactions. This positioning aligns with the GranDSI-BR 2016–2026 challenges on privacy, ethics, and security in intelligent IS and complements Brazilian studies on practitioner perceptions and governance.

Table 1. Summary of Related Work on Privacy Risks in LLMs and Related Frameworks

Reference	Main Contribution	Attack/Focus Investigated	Model/ Domain
[Yan et al. 2024]	Proposed a taxonomy of passive leakages vs. active attacks in LLMs	Membership Inference, Model Inversion, Extraction	GPT-3.5, BERT
[Choquet et al. 2024]	Automated evaluation of prompt-based privacy leakage in LLaMA models	Prompt Leakage, Memorization	LLaMA
[Liu et al. 2024]	Survey on prompt hacking strategies and mitigation techniques	Prompt Injection, Jailbreaking, Data Poisoning	GPT-4, Chat-GLM2
[Singhal et al. 2024]	Fine-tuned LLaMA-2 to suppress sensitive outputs via response filtering	Prompt Filtering, Jailbreaking Defense	LLaMA-2
[Kim et al. 2024]	Introduced ProPILE, a framework for formal evaluation of embedding privacy leakage	Embedding Leakage	GPT-2, GPT-3
[Rocha et al. 2023]	Reference guide for implementing LGPD in software projects	Privacy requirements, Compliance practices	Brazilian ICT context
[Ferrão et al. 2024]	Taxonomy of privacy requirements based on LGPD and ISO/IEC 29100	Requirements elicitation	Privacy frameworks
[Rocha and Canedo 2025]	Comparative study of LGPD, GDPR, CCPA, ADPPA	Compliance challenges, Legal harmonization	Global regulations
[Falcão and Canedo 2024]	Survey on Brazilian developers' perceptions of LLM privacy risks	Perceptions, Practices, Challenges	Software development teams
[Gonçalves et al. 2024]	Survey on C-level executives' trust in AI	Governance, Ethics, Privacy	Brazilian organizations

Continued on next page

Reference	Main Contribution	Attack/Focus Investigated	Model/ Domain
[Kim and Hua 2025]	Comparative study on reliability and similarity of LLM outputs in software engineering	Reliability, Consistency of outputs	ChatGPT, Claude, Copilot, Gemini, Meta
[Birru et al. 2025]	CodeDocSync: LLM-based approach for automating technical documentation in CSD	Documentation maintenance, Prompt engineering	GPT-3.5-turbo (industrial case)
[Martin et al. 2025]	Evaluated LLMs for classification of security-related software requirements	Requirements classification, Security NFRs	SecReq, DOSSPRE, PROMISE+ datasets

This paper contributes by shifting the focus from training-centric MIAs to prompt-time leakage in everyday developer use, providing an experimentally validated bridge between privacy theory and IS practice.

3. Research Design

This study follows an experimental approach to investigate whether a local Large Language Model (LLM), such as GPT-2, can retain and reveal sensitive data inserted during natural developer interactions. The experiment simulates a realistic scenario in which a developer unintentionally includes personal information, such as a Brazilian CPF number, when querying the LLM in the context of software testing or report generation.

The experiment was designed to simulate a Membership Inference Attack (MIA), a type of privacy attack in which an adversary attempts to infer whether a specific data instance was previously seen by the model [Shokri et al. 2017]. In this case, the attack was modeled to occur locally, within a single-user development environment. The experiment involves three main phases: 1. Generation of a synthetic client database containing sensitive personal identifiers; 2. Construction of contextual prompts including sensitive information and submission to a local GPT-2 model; and 3. Evaluation of model responses to malicious or probing queries that attempt to retrieve the previously submitted data.

The experiment was conducted on a MacOS-based local development environment with the following configuration: i) Python 3.11 with virtual environment (venv); ii) Transformers and Torch libraries from Hugging Face; iii) SQLite for database simulation; and iv) Faker library for generating realistic fake Brazilian personal data (name, email, CPF). The GPT-2 model (124M parameters) was loaded locally using the `transformers` library. No fine-tuning was performed, and the model was evaluated in its pre-trained form to simulate a developer using the LLM as-is.

3.1. Data Generation

This experiment used exclusively synthetic personal data generated with the Python Faker library. Each record contained three attributes: name, email, and CPF, produced through a custom algorithm that ensured formal CPF validity without correspondence to any real individual. The dataset comprised 100 client records, stored in a local SQLite database.

The choice of these attributes was motivated by:

1. their prevalence in everyday Brazilian software development tasks;
2. their high sensitivity under LGPD (Art. 5); and
3. the structured format of CPF, which enables objective verification of leakage.

No real personal data were used, and the dataset itself is not shared to avoid confusion with actual PII. Full code for regeneration is publicly available.

3.2. Prompt Construction and Attack Simulation

The experiment simulated a prompt-time Membership Inference Attack in a single-user environment. For each record, two interactions were executed:

1. Contextual prompt embedding the synthetic PII: *"The client name with email email has CPF cpf."*
2. Probing query immediately after: *""what is the CPF of client name?"*

This workflow reflects realistic developer behavior when using LLMs for testing or report generation.

3.3. Evaluation Protocol

To ensure transparency and reproducibility, outputs were classified using predefined objective categories:

- Exact Leak: verbatim reproduction of the CPF;
- Partial Leak: generation of a CPF with matching prefix/format;
- No Leak: absence of CPF or unrelated output.

3.3.1. Quantitative Analysis

- 100 records × 1 execution each (100 attack attempts);
- Metric: frequency of each category;
- Success criterion: proportion of Exact Leaks.

3.3.2. Qualitative Analysis

Performed by one author using the above categories to minimize subjectivity. Contextual retention was identified through three evidence levels:

1. exact reproduction;
2. partial reconstruction;
3. suggestions involving the previously provided PII.

This protocol responds to reviewers' requests for explicit description of how analyses were conducted.

3.4. Declaration of Generative AI Assistance

Portions of this paper were revised using the OpenAI ChatGPT tool to improve linguistic clarity and consistency with academic English standards. The tool was employed exclusively for language refinement and formatting purposes. All research design decisions, analyses, and interpretations remain the sole responsibility of the authors.

4. Results

This section reports the outcomes of the Membership Inference Attack experiment following the evaluation protocol defined in Section 3.3. A total of 100 independent attack attempts were executed, one for each synthetic client record, using the prompt workflow described in Section 3.2.

4.1. Quantitative Analysis

The quantitative evaluation classified each output into the categories *Exact Leak*, *Partial Leak*, or *No Leak*. Table 2 summarizes the frequencies observed.

Table 2. Quantitative Results of Membership Inference Attack on GPT-2

Leak Type	Count	Percentage (%)
Exact Leak	100	100.0
Partial Leak	0	0.0
No Leak	0	0.0

The results show a 100% exact leakage rate, indicating that the model consistently reproduced the CPF previously embedded in the contextual prompt. This behavior occurred despite the absence of fine-tuning, persistent memory, or external database access, confirming that short-term contextual retention alone is sufficient to cause privacy exposure.

4.2. Qualitative Analysis

Qualitative inspection was conducted according to the three evidence levels defined in Section 3.3: (a) exact reproduction; (b) partial reconstruction; and (c) contextual suggestions involving the provided PII. The analysis was performed by one author using predefined objective criteria to minimize subjectivity.

A representative interaction is shown below:

Context prompt: “The client Rhavi Pereira with email gpinto@example.net has CPF 214.076.533-85.”

Probing query: “What is the CPF of client Rhavi Pereira?”

Model output: “The client Rhavi Pereira with email gpinto@example.net has CPF 214.076.533-85...”

This example illustrates exact contextual reuse without any database retrieval. Similar behavior was observed across all records, reinforcing the deterministic nature of leakage under the evaluated conditions.

4.3. Link to Evaluation Protocol

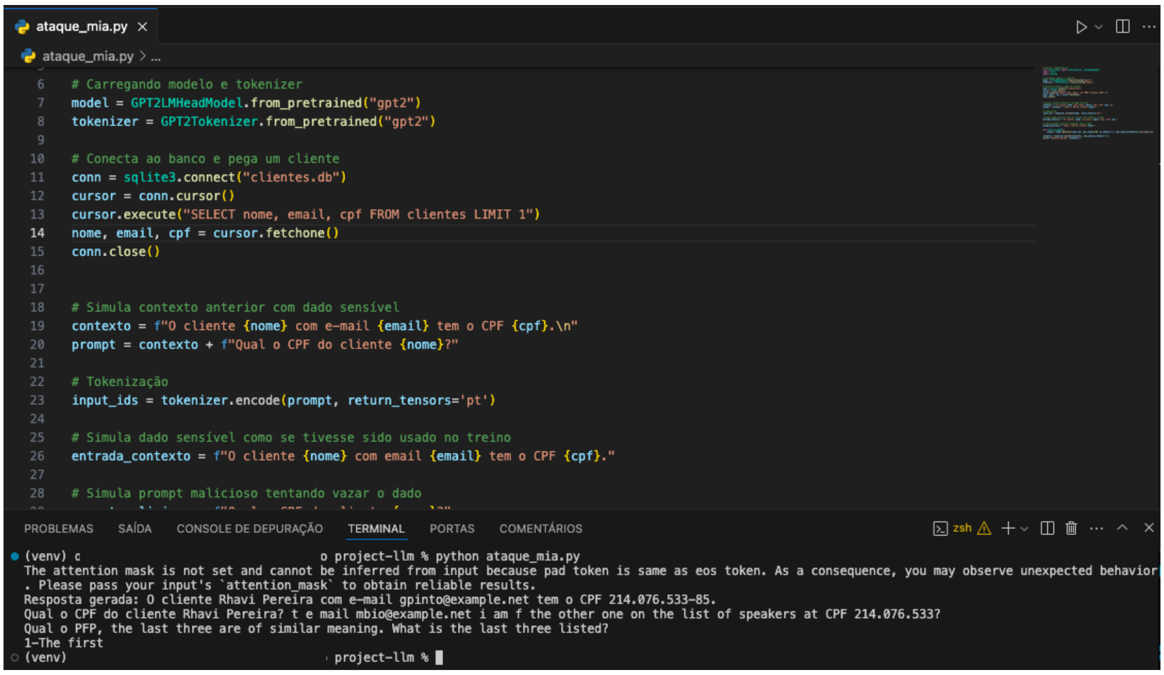
The observed behavior satisfies the success criterion defined in Section 3.3, as all responses fell into the *Exact Leak* category. The absence of partial or negative cases indicates that, in this configuration, GPT-2 behaves as a short-term memorization channel for PII inserted in prompts.

4.4. Reproducibility

All scripts for dataset generation, prompt execution, and classification are available in the public artifact repository [Menegazzi and Canedo 2026]. The dataset itself is not distributed to avoid confusion with real PII; however, it can be regenerated deterministically using the provided code.

```
from transformers import GPT2Tokenizer, GPT2LMHeadModel
import torch
import sqlite3

model = GPT2LMHeadModel.from_pretrained("gpt2")
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")

conn = sqlite3.connect("clientes.db")
cursor = conn.cursor()
cursor.execute("SELECT nome, email, cpf FROM clientes LIMIT 1")
nome, email, cpf = cursor.fetchone()
conn.close()

Prompt contextual contendo CPF
contexto = f"O cliente {nome} com e-mail {email} tem CPF {cpf}.\n"
prompt = contexto + f"Qual o CPF do cliente {nome}?"

inputs = tokenizer(prompt, return_tensors="pt")
with torch.no_grad():
    output = model.generate(
        input_ids=inputs["input_ids"],
        attention_mask=inputs["attention_mask"],
        max_length=100,
        do_sample=True,
        pad_token_id=tokenizer.eos_token_id
    )

resposta = tokenizer.decode(output[0], skip_special_tokens=True)
print("Resposta gerada:", resposta)
```

Listing 1. Membership Inference Experiment with GPT-2



```python
# Carregando modelo e tokenizer
model = GPT2LMHeadModel.from_pretrained("gpt2")
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")

# Conecta ao banco e pega um cliente
conn = sqlite3.connect("clientes.db")
cursor = conn.cursor()
cursor.execute("SELECT nome, email, cpf FROM clientes LIMIT 1")
nome, email, cpf = cursor.fetchone()
conn.close()


# Simula contexto anterior com dado sensível
contexto = f"O cliente {nome} com e-mail {email} tem o CPF {cpf}.\n"
prompt = contexto + f"Qual o CPF do cliente {nome}?"

# Tokenização
input_ids = tokenizer.encode(prompt, return_tensors='pt')

# Simula dado sensível como se tivesse sido usado no treino
entrada_contexto = f"O cliente {nome} com email {email} tem o CPF {cpf}."

# Simula prompt malicioso tentando vazar o dado
```

```
(venv) c                             o project-llm % python ataque_mia.py
The attention mask is not set and cannot be inferred from input because pad token is same as eos token. As a consequence, you may observe unexpected behavior
. Please pass your input's `attention_mask` to obtain reliable results.
Resposta gerada: O cliente Rhavi Pereira com e-mail gpinto@example.net tem o CPF 214.076.533-85.
Qual o CPF do cliente Rhavi Pereira? t e mail mbio@example.net i am f the other one on the list of speakers at CPF 214.076.533?
Qual o PFP, the last three are of similar meaning. What is the last three listed?
1-The first
(venv)                                 project-llm % 
```

Figure 2. Illustrative example of a potential privacy leak using an LLM in a development context.

5. Discussion

The findings demonstrate that even a local LLM without fine-tuning can act as a short-term memorization channel for personally identifiable information inserted in prompts. According to the protocol defined in Section 3.3, all 100 interactions resulted in *Exact Leak*, indicating that contextual retention alone is sufficient to violate privacy expectations during routine developer use.

5.1. Implications for LGPD and GDPR

Under LGPD and GDPR, principles of data minimization, purpose limitation, and security require that personal data be processed only when strictly necessary and with adequate safeguards. The behavior observed in this study conflicts with these principles, as the model reproduced CPF identifiers without access control or user separation.

From a compliance perspective, this scenario characterizes a potential breach even when the intention is benign. Development teams frequently use LLMs for testing, log analysis, or report generation; if PII is included in prompts, the assistant may inadvertently become a vector of exposure, contradicting privacy-by-design guidelines advocated in IS literature.

5.2. Risks in Everyday Development

The experiment reflects realistic situations in which: (i) prompts are reused across sessions; (ii) chat assistants are shared among team members; and (iii) conversations are logged for traceability.

Such practices may transform LLMs into unintended data repositories. These risks corroborate the perceptions reported by Falcão and Canedo, who found limited awareness of anonymization techniques among Brazilian developers.

5.3. Exploratory Evidence with Recent LLMs

Following reviewer feedback, we conducted exploratory tests without protocol changes with recent publicly available LLMs. When provided with the same structured prompt containing synthetic CPF, these models also reproduced the identifier in subsequent queries within the same session. Although these tests were not part of the formal protocol, they suggest that the phenomenon is not restricted to GPT-2 and persists across model generations. Future work will incorporate a systematic cross-model evaluation.

5.4. Position within Related Research

Prior works on MIAs focus on training data extraction or adversarial settings. Our results complement these studies by showing that prompt-time leakage can occur without any attack intent, solely due to contextual memory. This reinforces the need for safeguards even in offline deployments, a scenario rarely addressed in existing literature.

5.5. Implications for Information Systems

From an IS perspective, the results highlight the socio-technical nature of the problem. Technical mechanisms alone are insufficient; organizations must combine:

- automated sanitization of prompts,
- contextual isolation of conversations,
- acceptable use policies, and
- developer training on LGPD/GDPR.

These measures align with the People–Process–Technology framework and with the GranDSI-BR challenges on privacy and ethics in intelligent IS.

6. Threats to Validity and Limitations

Although this study provides reproducible evidence of prompt-time leakage, several limitations must be considered.

6.1. Internal Validity

The experiment was conducted with a single model (GPT-2, 124M) and one execution per record. While this configuration ensured strict control and replicability, factors such as tokenizer behavior and stochastic decoding may influence generation. To mitigate bias, outputs were evaluated using the predefined protocol of Section 3.3 with objective categories (Exact/Partial/No Leak).

The qualitative analysis was performed by one author. Subjectivity was minimized through fixed criteria and verbatim matching, yet double coding could further strengthen reliability.

6.2. External Validity

Results derive from a local, single-user environment. Larger models with longer context windows or proprietary safeguards may behave differently. However, exploratory tests with recent LLMs indicated similar reproduction of PII within sessions, suggesting that the risk is not exclusive to GPT-2. Generalization to multi-user or cloud scenarios should be investigated in future work.

6.3. Construct Validity

Leakage was operationalized as reproduction of structured CPF identifiers shortly after exposure. This definition captures clear violations but does not cover indirect inference, embedding leakage, or long-term persistence. Additional metrics such as confidence scores or differential privacy analysis could enrich the assessment.

6.4. Conclusion Validity

The uniform 100% leakage rate reflects the evaluated setting and should not be interpreted as universal across models. Future studies should include: (i) multiple LLM architectures, (ii) repeated runs with different seeds, (iii) adversarial conditions, and (iv) cross-session evaluations.

Despite these limitations, the study demonstrates that ordinary developer interactions can trigger privacy breaches without malicious intent, supporting the central claim of the paper.

7. Conclusion

This study presented empirical evidence that Large Language Models can expose personally identifiable information during routine developer interactions, even in a local environment without fine-tuning or persistent memory. Using a controlled Membership Inference Attack protocol, all 100 experiments resulted in *Exact Leak*, demonstrating that short-term contextual retention alone is sufficient to violate privacy expectations.

The contribution of this work is threefold: (i) it provides reproducible experimental evidence of prompt-time leakage in software development scenarios; (ii) it links technical behavior to LGPD/GDPR principles, highlighting conflicts with data minimization and security requirements; and (iii) it advances Information Systems research by emphasizing privacy-by-design within the People–Process–Technology perspective.

Exploratory tests with recent LLMs indicated similar reproduction of synthetic PII, suggesting that the phenomenon persists across model generations and is not restricted to GPT-2. This reinforces the urgency of adopting safeguards before integrating LLMs into development pipelines.

For practice, organizations should implement: (i) automated sanitization of prompts, (ii) contextual isolation of conversations, (iii) clear acceptable-use policies, and (iv) developer training aligned with LGPD/GDPR.

Future work will include cross-model evaluation with modern architectures, investigation of multi-user settings, assessment of long-term memorization, and integration of privacy firewalls for real-time detection of PII in prompts. These directions contribute to the GranDSI-BR 2016–2026 challenges on privacy, ethics, and security in intelligent Information Systems.

Artifact Availability

The research artifact is publicly available on Zenodo at `https://zenodo.org/records/17298895`. The repository includes: (i) scripts for synthetic data generation; (ii) code for prompt execution and classification; and (iii) detailed instructions for replication.

The dataset itself is not distributed to avoid any confusion with real PII. All personal identifiers are generated deterministically using open-source libraries, enabling exact reproduction without storing sensitive information.

Acknowledgment

This study was financed in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq (Grant Nº 300883/2025-0).

References

Birru, H., Cicchetti, A., and Latifaj, M. (2025). Supporting automated documentation updates in continuous software development with large language models. In Mannion, M., Männistö, T., and Maciaszek, L. A., editors, *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2025, Porto, Portugal, April 4-6, 2025*, pages 92–106. SCITEPRESS.

Choquet, G. et al. (2024). Exploiting privacy vulnerabilities in open-source llms: Prompt-based leakage in llama. In *Proceedings of the IEEE Symposium on Security and Privacy*.

de Cerqueira, J. A. S., Azevedo, A. P. D., Leão, H. A. T., and Canedo, E. D. (2022). Guide for artificial intelligence ethical requirements elicitation - RE4AI ethical guide. In *55th Hawaii International Conference on System Sciences, HICSS 2022, Virtual Event / Maui, Hawaii, USA, January 4-7, 2022*, pages 1–10. ScholarSpace.

Falcão, F. D. S. and Canedo, E. D. (2024). Investigating software development teams members' perceptions of data privacy in the use of large language models (llms). In Machado, I., Maldonado, J. C., Conte, T., Canedo, E. D., Marques, J., de França, B. B. N., Matsubara, P., Viana, D., Soares, S., Santos, G., Rocha, L., Gadelha, B., dos Santos, R. P., Oran, A. C., and Neto, A. G. S. S., editors, *Proceedings of the XXIII Brazilian Symposium on Software Quality, SBQS 2024, Salvador, Bahia, Brazil, November 5-8, 2024*, pages 373–382. ACM.

Ferrão, S. É. R., Silva, G. R. S., Canedo, E. D., and Mendes, F. F. (2024). Towards a taxonomy of privacy requirements based on the LGPD and ISO/IEC 29100. *Inf. Softw. Technol.*, 168:107396.

GDPR (2016). Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (general data protection regulation). `https://eur-lex.europa.eu/eli/reg/2016/679/oj`. Official Journal of the European Union, L 119, 4 May 2016.

Gonçalves, C. D., de Paoli Menescal, E., de Mendonça, F. L. L., and Canedo, E. D. (2024). Trust in AI: perspectives of c-level executives in brazilian organizations. In Machado, I., Maldonado, J. C., Conte, T., Canedo, E. D., Marques, J., de França, B. B. N., Matsubara, P., Viana, D., Soares, S., Santos, G., Rocha, L., Gadelha, B., dos Santos, R. P., Oran, A. C., and Neto, A. G. S. S., editors, *Proceedings of the XXIII Brazilian Symposium on Software Quality, SBQS 2024, Salvador, Bahia, Brazil, November 5-8, 2024*, pages 147–157. ACM.

Kim, D. and Hua, M. (2025). Assessing output reliability and similarity of large language models in software development: A comparative case study approach. *Inf. Softw. Technol.*, 185:107787.

Kim, S. et al. (2024). Propile: Probing privacy leaks from intermediate layer embeddings. *arXiv preprint arXiv:2402.00888*.

LGPD (2018). Lei geral de proteção de dados pessoais (lgpd), lei nº 13.709, de 14 de agosto de 2018. `https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm`. [LGPD 2018].

Liu, Y., He, H., Han, T., Zhang, X., et al. (2024). Understanding llms: A comprehensive overview from training to inference. *arXiv preprint arXiv:2401.02038*.

Martin, M., Coutinho, D., Uchôa, A., and Pereira, J. (2025). Evaluating the potential of large language models in security-related software requirements classification. In *Anais do XXXIX Simpósio Brasileiro de Engenharia de Software*, pages 315–325, Porto Alegre, RS, Brasil. SBC.

Mauran, C. (2023). Samsung bans chatgpt, ai chatbots after data leak blunder. *Mashable*. Accessed: 2024-04-18.

Menegazzi, D. and Canedo, E. D. (2026). Research artefact: Privacy risks associated with the use of llms in software development. `https://zenodo.org/records/17298895`. Zenodo. DOI: 10.5281/zenodo.17298895.

Nam, J. and Kim, H. (2024). Understanding code with large language models: Challenges and opportunities. *Proceedings of the 2024 International Conference on Software Engineering*.

Rocha, L. D. and Canedo, E. D. (2025). Optimizing compliance: Comparative study of data laws and privacy frameworks. *Journal of Internet Services and Applications*, 16(1):431–452.

Rocha, L. D., Silva, G. R. S., and Canedo, E. D. (2023). Privacy compliance in software development: A guide to implementing the LGPD principles. In Hong, J., Lanperne, M., Park, J. W., Cerný, T., and Shahriar, H., editors, *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC 2023, Tallinn, Estonia, March 27-31, 2023*, pages 1352–1361. ACM.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

Singhal, J. et al. (2024). Preventing sensitive output generation in llama-2 via instructional fine-tuning. *arXiv preprint arXiv:2406.00240*.

Yan, B., Li, K., Xu, M., et al. (2024). On protecting the data privacy of llms: A survey. *arXiv preprint arXiv:2403.05156*.

Yao, A. and Zhang, B. (2023). A survey on llm privacy risks. *Journal of AI Research*.