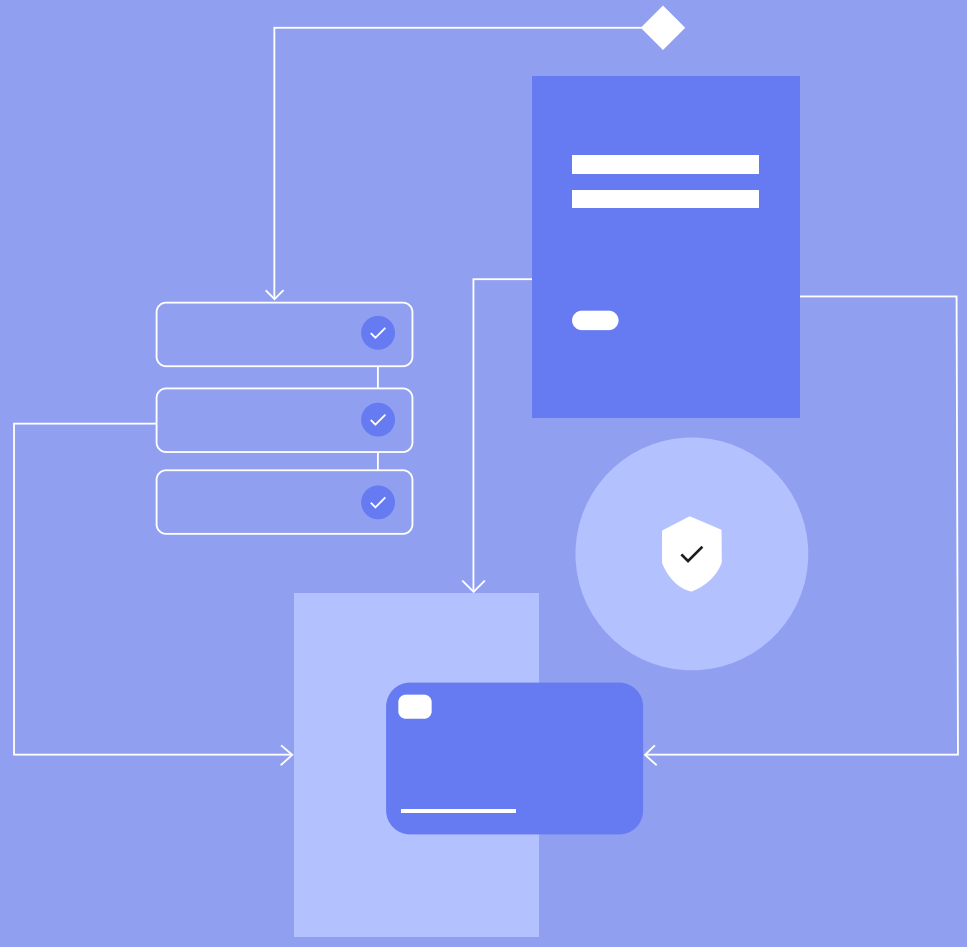
Security in Mission Critical Applications: Case ‘Uruguay gets vaccinated’

Whitepaper



GeneXus™

On March 1, 2021, the first stage of the COVID-19 Vaccination Plan was activated in Uruguay, under the slogan #Uruguayseva-cuna.

As vaccines were limited at that time, government authorities decided to prioritize at-risk groups, consisting of patients with morbidities and workers who could be more exposed to contracting the virus, such as health and education personnel, police officers, firefighters, among others. They would then move on to the rest of the population.

For the vaccination to take place in an orderly and safe manner, a digital agenda was created with GeneXus, accessible from the web, from the CoronavirusUY mobile application (Google Play - App Store) and from a chat.

The system had to be developed in record time. Positive cases were increasing, as was the mortality rate related to this disease. Having vaccinations and not being able to administer them due to system failures was not an option. That is why the software developed with Low-Code for #Uruguaysevacuna is considered a **Mission Critical Application.**

So it was that in just 2 weeks, a multidisciplinary team created the first version of this solution that allowed 2 million people, over 18 years of age, and eligible to receive the COVID-19 vaccine, to request their schedule, get vaccinated at the assigned time, place and date, and then be notified to apply the second dose.

"The system had to support a high level of transactions. What was expected was that at specific moments we would receive a lot of requests at the same time, then this number of requests would go down but could go up in the face of different events and generate peaks. Our challenge consisted of developing a secure system in accordance with the legislation of the National Integrated Health System of the Eastern Republic of Uruguay," explains Gerardo Canedo, Computer Engineer, IT Security Specialist and Security Manager at GeneXus Consulting.

People, Ideas, Tools. What's Behind Coronavirus UY?

Read more

A risk-oriented approach

The team:

✓ This system was built by people working in remote mode.

✓ The security issue fell on everyone, from business analysts, to testers, developers, and software architects.

The strategy:

They had to be effective and efficient. To achieve this, and long before starting the coding process, they focused on IT security, identifying:

✓ The major risks they had to mitigate.

✓ The risks they were not going to mitigate but had to know how they were going to manage them.

Tasks:

"Of all the activities we could do with security in mind, we decided in this first iteration to take into account the four with the most added value, which are Threat Modeling, Architecture Risk Analysis, Security Requirements Definition and Security Testing," Canedo details.

Threat Modeling

This is the process by which threats are identified, the vulnerabilities that may exist, and how these vulnerabilities could be used to carry out an attack.

For this, they answered the following questions:

-Who might be interested in attacking this system?

-What could be the target(s) of the attack?

-What type of attackers could the system have?

-What techniques might be used to attack the system and how to resolve it?

"We had to implement controls to mitigate or reduce the impact capability or potential for a successful attack. In the Threat Modeling game, we sought to understand who would be on the other side trying to damage the system."

The following conclusions were drawn from this analysis:

✓ The system had to be restricted to the national territory.

✓ Anti-automation mechanisms had to be implemented for public interfaces to the world.

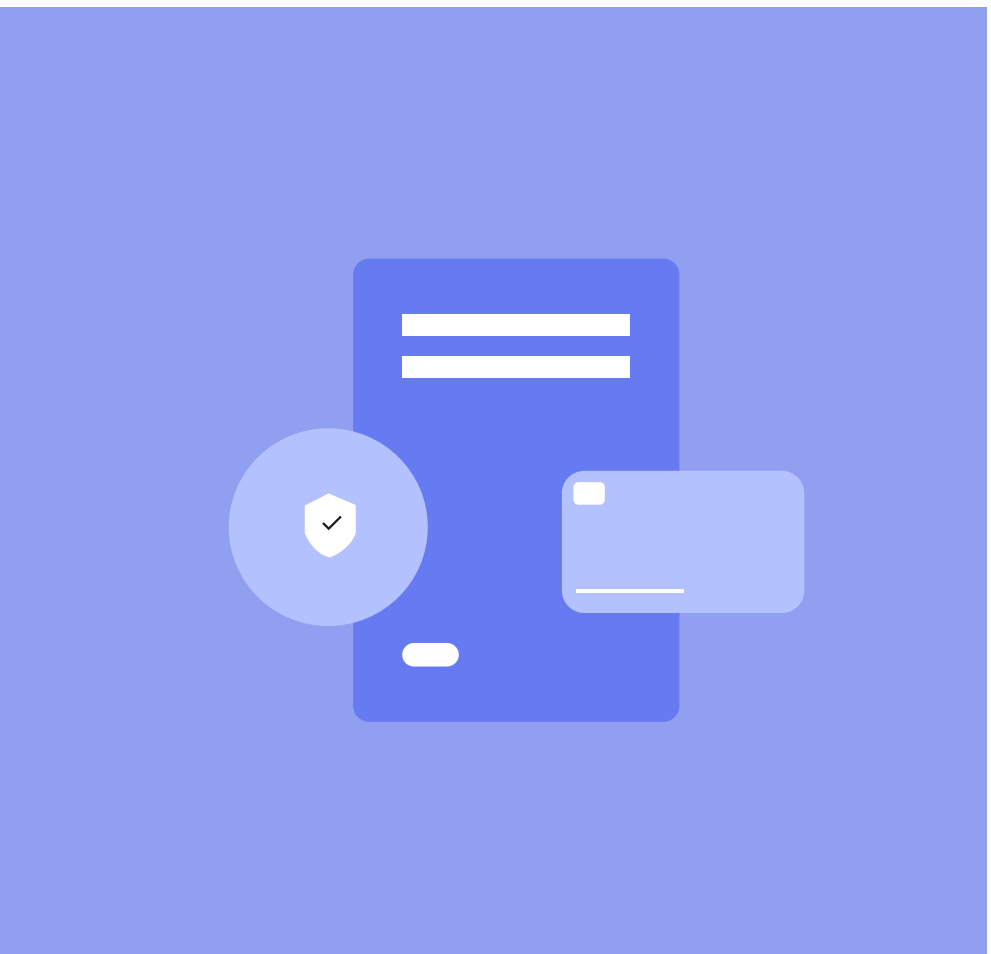
✓ They could not store personal data in the cloud they were going to use, since this cloud is not located in Uruguay.

✓ To identify, register and continuously monitor a person, they had to request the date of birth and ID number.

Risk and Architecture Analysis

In this task they evaluated the proposed architecture from a security standpoint, identifying areas of concern and how attacks could be remedied.

"What we had as input was a high-level architecture diagram, in which there was not the complete document (because it was not yet assembled), but we did have an idea of how the architecture of this solution was going to be. From there we identified the information flows and defined the system boundaries, the trust zones (which were two), the flows and the connections. We had to make sure that the communication between these trust zones was secure. For that, we used the STRIDE methodology, identifying possible vulnerabilities and how we could mitigate them", explains the also member of the Uruguayan chapter of the Open Web **Application Security Projects (OWASP).**



The problem: the public cloud

"One of these areas is the public cloud that is outside the national territory. To comply with the country's requirements and regulations, we could not store personal information there and we had to handle as little data as possible. That implied a design and architecture challenge. We had to store information in the cloud without storing personal data, and we still had to be able to determine whether or not a person was eligible to be vaccinated."

The solution

"We put together an improvement to the architecture, defining a way to store the personal data information. For this, we used some cryptographic mechanisms, such as the hash function.

Thanks to this refined architecture, we were able to create a document containing the risk analysis of each of the components and the threat modeling, how they could be executed and the motivations that attackers could have in each case."

Defining Security Requirements

The security requirements emerged from the previous tasks and were captured in a document that was shared with the entire team.

Security testing

"One of the most effective controls we implemented was data type **validation in the APIs.** Each of the APIs' data inputs were varied with respect to certain regular expressions. By performing those validations, we very efficiently looked at many attacks that could occur in other contexts.

Another option we implemented was the definition of **Abuse Cases**. Basically, we hacked the application on paper, putting together how an adversary would have to attack in order to damage the system. With that information, we started to perform security tests every time a component was released. And this is where the abuse cases came into use. We took those cases that we had seen that could not happen and proved that they did not happen in the system.

From that information, we were able to prove that the risks that mattered most to us could not reasonably be realized. As a result, we were able to come out with the first version of the system in the planned time, with a known level of security, with mitigated risks and assumed risks".

For Canedo, it is utopian to believe that a system can be risk-free. However, this process allowed them to:

- ✓ Know the security level of the application.
- ✓ To know the strengths and weaknesses of the system.
- ✓ Determine the controls they had, and how to proceed in case of an attack.

In Video

To learn more about this topic, we invite you to watch the talk Application Security in Mission-critical Software: A Proactive and Cost Effective Approach, given by Canedo at the last edition of GeneXus Live.

Conceptualizing an API for the app CoronavirusUy

In Uruguay, when the pandemic began in March 2020, it was planned through the National Coronavirus Plan, the possibility of offering different methods of contact between citizens and health care providers, to coordinate COVID-19 tests. Thus, the CoronavirusUy application and chats through the Ministry of Public Health website, Facebook Messenger and WhatsApp were created.

In 2021, when the digital agenda process to get vaccinated against COVID-19 began, it was defined to use the same communication channels. The solution became a Mission Critical Application, because of the number of people who were going to try to schedule at the same time.

"We had to think of a completely different architecture. We needed a mechanism, an asynchronous architecture where all these people, accessing from different applications and different interfaces, could schedule themselves and that this would not saturate the systems that already existed at the Ministry of Public Health level. What we did was to conceptualize an API, separating the process of scheduling from the process of checking whether the person was already scheduled and what status they were in," explains Eugenio García, Product Manager of GeneXus For SAP Systems.

The objective? To allow all people to be scheduled without saturating the system.
"For this purpose, a mediation layer was created, where this information was, in turn, stored in the Amazon SQS infrastructure. Then, through GeneXus, a Business Logic layer was built to read the information and gradually pass it to the electronic agenda system. In this way, each time it was confirmed, the status of each of these agendas was stored in a DynamoB data structure, which is more suitable for these systems, where great scalability is also required at the reading level".

If you want to know more, be sure to watch the talk Innovating in the API Economy with GeneXus, given by Eugenio García as part of GeneXus LIVE. In the presentation he also explains how to work with GeneXus in different integration scenarios: on the one hand bringing information from a third party API and on the other hand exposing with GeneXus the API that others can use.

GeneXus™

MONTEVIDEO - URUGUAY	Av. Italia 6201- Edif. Los Pinos, P1	(598) 2601 2082
CIUDAD DE MÉXICO - MÉXICO	Hegel N° 221, Piso 2, Polanco V Secc.	(52) 55 5255 4733
MIAMI - USA	8950 SW 74th Ct, Suite 1406	(1) 201 603 2022
SÃO PAULO - BRASIL	Rua Samuel Morse 120 Conj. 141	(55) 11 4858 0300
TOKYO - JAPAN	2-27-3, Nishi-Gotanda	(81) 3 6303 9381
	Shinagawa-ku, Tokyo, 141-0031	(81) 3 6303 9980