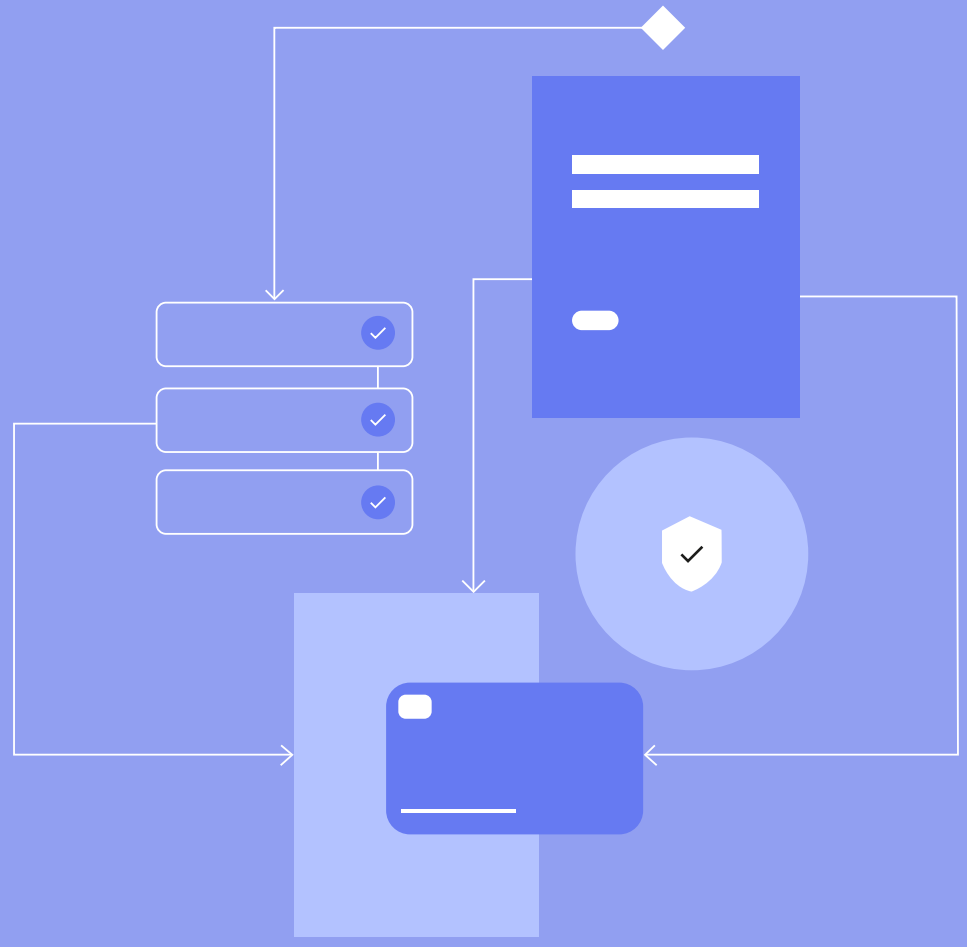
Segurança em aplicações de Missão Crítica: O caso ‘Uruguay se Vacuna’

Whitepaper



GeneXus™

Em 1º de março de 2021, a primeira etapa do Plano de Vacinação COVID-19 foi ativada no Uruguai, sob o slogan #Uruguaysevacuna.

Como as vacinas eram limitadas naquela época, as autoridades governamentais decidiram dar prioridade aos grupos de risco, constituídos por pacientes com comorbidades e trabalhadores que poderiam estar mais expostos ao vírus, tais como pessoal de saúde e educação, policiais, bombeiros, entre outros. Após, eles passariam então para o resto da população.

Para que a vacinação ocorresse de forma ordenada e segura, foi criada uma agenda digital com GeneXus, acessível pela web, a partir do aplicativo mobile CoronavirusUY (Google Play - App Store) e de um chat.

O sistema teve que ser desenvolvido em tempo recorde. Os casos positivos estavam aumentando, assim como a taxa de mortalidade relacionada a esta doença. Ter vacinações e não poder administrá-las devido a falhas no sistema não era uma opção. É por isso que o software desenvolvido com Low-Code para #Uruguaysevacuna é considerado uma **Aplicação de Missão Crítica.**

Assim, em apenas 2 semanas, uma equipe multidisciplinar criou a primeira versão desta solução que permitiu que 2 milhões de pessoas, maiores de 18 anos e elegíveis para receber a vacina COVID-19, solicitassem sua programação, fossem vacinadas no horário, local e data designados e depois fossem notificadas para a segunda dose.

"O sistema tinha que suportar um alto nível de transações. A expectativa era de que haveria muitos pedidos de uma só vez, então o número de pedidos diminuiria, mas poderia subir com diferentes eventos e gerar picos. Nosso desafio era desenvolver um sistema seguro de acordo com a legislação do Sistema Nacional Integrado de Saúde da República Oriental do Uruguai", explica Gerardo Canedo, Engenheiro de Informática, Especialista em Segurança TI e Gerente de Segurança da GeneXus Consulting.

Pessoas, ideias, ferramentas... o que está por trás do Coronavirus UY?

Leia mais

Uma abordagem orientada ao risco

A equipe:

✓ Este sistema foi construído por pessoas que trabalham remotamente.

✓ A questão da segurança recaiu sobre todos, desde analistas de negócios, até testadores, desenvolvedores e arquitetos de software.

A estratégia:

Eles tinham que ser eficazes e eficientes. Para conseguir isso, muito antes de iniciar o processo de codificação, eles se concentraram na segurança de TI, identificando:

✓ Os maiores riscos que eles tinham que mitigar.

✓ Os riscos que eles não iriam mitigar, mas tinham que saber como iriam gerenciá-los.

As tarefas:

"De todas as atividades que podíamos fazer com a segurança em mente, decidimos nesta primeira iteração levar em conta as quatro de maior valor agregado, que são: Modelagem de Ameaças, Análise de Risco de Arquitetura, Definição de Requisitos de Segurança e Testes de Segurança", detalha Canedo.

Modelagem de Ameaças:

É o processo pelo qual as ameaças são identificadas, as vulnerabilidades que podem existir e como essas vulnerabilidades podem ser usadas para realizar um ataque.

Para isso, eles responderam às seguintes perguntas:

-Quem poderia estar interessado em atacar este sistema?

-Qual poderia ser o(s) alvo(s) do ataque?

-Qual tipo de ataques o sistema poderia ter?

Quais técnicas poderiam ser usadas para atacar o sistema e como resolvê-lo?

"Tivemos que implementar controles para mitigar ou reduzir o impacto ou o potencial para um ataque bem sucedido". No jogo de Modelagem de Ameaças, procuramos entender quem estaria do outro lado tentando danificar o sistema".

As seguintes conclusões foram tiradas desta análise:

✓ O sistema tinha que ser restrito ao território nacional.

✓ Mecanismos anti-automação tiveram que ser implementados para interfaces públicas com o mundo.

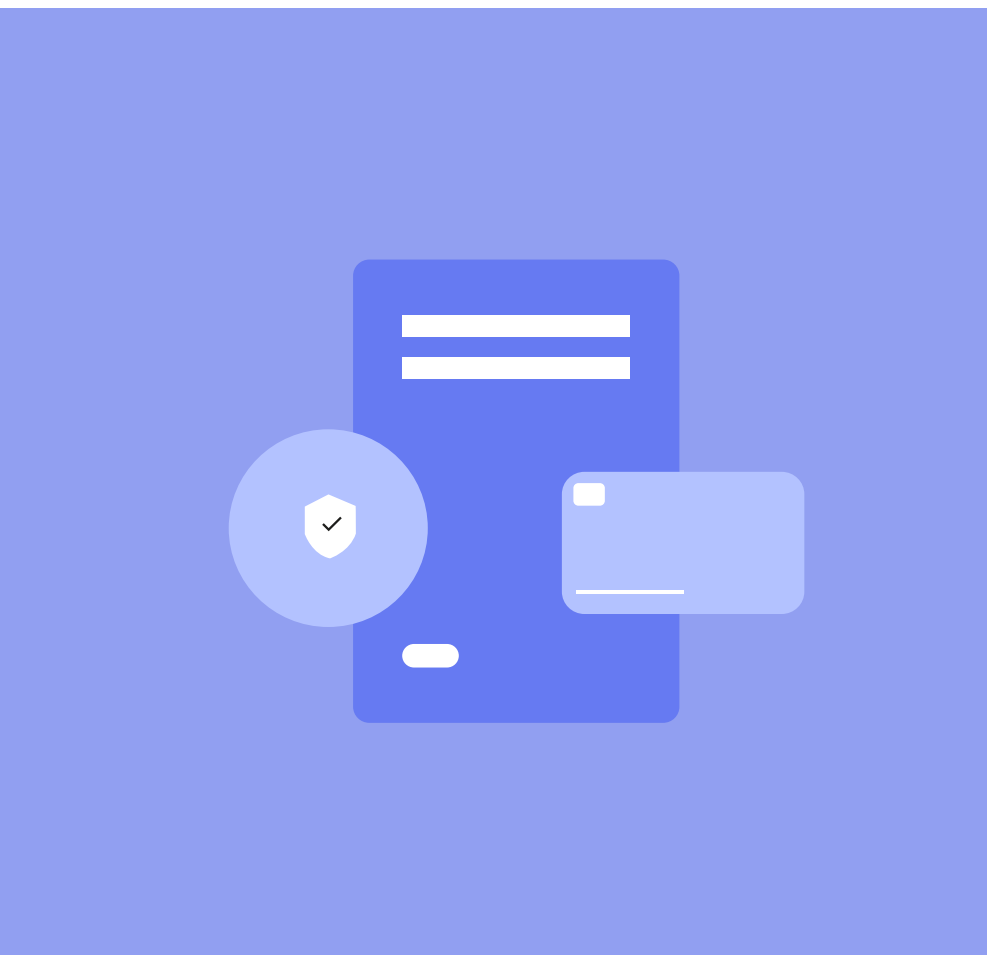
✓ Eles não podiam armazenar dados pessoais na nuvem que iriam utilizar, pois estas nuvens não estão localizadas no Uruguai.

✓ Para identificar, registrar e monitorar continuamente uma pessoa, eles tinham que solicitar a data de nascimento e o número de identificação.

Análise de Risco e Arquitetura:

Nesta tarefa, eles avaliaram a arquitetura proposta do ponto de vista da segurança, identificando áreas de preocupação e como os ataques poderiam ser tratados.

"O que tínhamos como entrada era um diagrama de arquitetura de alto nível, que não era o documento completo (porque ainda não estava montado), mas tínhamos uma ideia de como seria a arquitetura desta solução. A partir daí identificamos os fluxos de informação e definimos os limites do sistema, as zonas de confiança (das quais existiam duas), os fluxos e as conexões. Tínhamos que nos certificar de que a comunicação entre essas zonas de confiança fosse segura. Para isso utilizamos a metodologia STRIDE, identificando as possíveis vulnerabilidades e como poderíamos mitigá-las", explica o também membro do capítulo uruguaio do **Open Web Application Security Projects (OWASP).**



O problema: a nuvem pública

"Uma dessas áreas é a nuvem pública, que está fora do território nacional. A fim de cumprir as exigências e regulamentos do país, não podíamos armazenar informações pessoais lá e tínhamos que tratar o mínimo possível de dados. Isso implicava um desafio de projeto e arquitetura. Tivemos que armazenar informações na nuvem sem armazenar dados pessoais e ainda ser capazes de determinar se uma pessoa era ou não realmente elegível para a vacinação.

A solução

"Construímos uma arquitetura melhorada, definindo uma forma de armazenar as informações de dados pessoais. Para isso, usamos alguns mecanismos criptográficos, como a função hash.

Graças a esta arquitetura refinada, pudemos criar um documento contendo a análise de risco de cada um dos componentes e a modelagem da ameaça, como eles poderiam ser executados e as motivações que os invasores poderiam ter em cada caso.

Definindo os requisitos de segurança:

Os requisitos de segurança emergiram das tarefas anteriores e foram capturados em um documento que foi compartilhado com toda a equipe.

Teste de segurança:

"Um dos controles mais eficazes que implementamos foi a **validação do tipo de dados nas APIs**. Cada uma das entradas de dados API foi variada com respeito a certas expressões regulares. Ao realizarmos estas validações, analisamos com muita eficiência muitos ataques que poderiam ocorrer em outros contextos.

Outra opção que implementamos foi a definição de casos de abuso. Basicamente, nós hackeamos a aplicação no papel, reunindo como um adversário teria que atacar a fim de danificar o sistema. Com essas informações, começamos a realizar testes de segurança toda vez que um componente era liberado. E foi aqui que os casos de abuso entraram em uso. Pegamos os casos que tínhamos visto que não poderiam acontecer e provamos que eles não aconteciam no sistema.

A partir dessas informações, pudemos provar que os riscos que mais nos interessavam não podiam ser razoavelmente realizados. Como resultado, conseguimos apresentar a primeira versão do sistema no tempo planejado, com um nível de segurança conhecido, com riscos mitigados e riscos assumidos".

Para Canedo, é utópico acreditar que um sistema pode ser livre de riscos. No entanto, este processo permitiu que o fizessem:

✓ Conhecer o nível de segurança da aplicação.

✓ Conhecer os pontos fortes e fracos do sistema.

✓ Determinar que controles tinham, e como proceder em caso de um ataque.

Video

Para saber mais sobre este tópico, convidamos você a assistir à palestra Segurança de Aplicativos em Software de Missão Crítica: Uma abordagem proativa e lucrativa, dada pelo Canedo na última edição do GeneXus Live.

Conceitualizando uma API para o aplicativo CoronavirusUy

No Uruguai, quando a pandemia começou em março de 2020, o Plano Nacional de Coronavírus planejava oferecer diferentes métodos de contato entre cidadãos e prestadores de serviços de saúde para coordenar os testes COVID-19. Isto levou à criação do aplicativo CoronavirusUy e de chats através do site do Ministério da Saúde Pública, Facebook Messenger e WhatsApp.

Em 2021, quando o processo de agenda digital para vacinação contra a COVID-19 começou, foi decidido utilizar os mesmos canais de comunicação. A solução tornou-se uma Aplicação de Missão Crítica, devido ao número de pessoas que iriam tentar agendar ao mesmo tempo.

"Tivemos que inventar uma arquitetura completamente diferente. Precisávamos de um mecanismo, uma arquitetura assíncrona onde todas essas pessoas, acessando a partir de diferentes aplicações e diferentes interfaces, pudessem se programar e que isso não saturasse os sistemas que já existiam no nível do Ministério da Saúde Pública. O que fizemos foi conceitualizar um API, separando o processo de agendamento do processo de verificação se a pessoa já estava agendada e em que estado estava", explica Eugenio García, Gerente de Produto da GeneXus For SAP Systems.

O objetivo? permitir que todas as pessoas possam ser programadas sem saturar o sistema. "Para isto, foi criada uma camada de mediação, onde estas informações foram, por sua vez, armazenadas na infra-estrutura SQS da Amazônia. Então, através do GeneXus, uma camada de Lógica de Negócios foi construída para ler a informação e passá-la gradualmente para o sistema de agenda eletrônica. Desta forma, cada vez que foi confirmado, o status de cada uma destas agendas foi armazenado em uma estrutura de dados DynamoB, que é mais adequada para estes sistemas, onde também é necessária uma grande escalabilidade no nível de leitura.

Se você quiser saber mais, não perca a palestra Inovando na Economia API com GeneXus, dada por Eugenio García no âmbito do GeneXus LIVE. Na apresentação ele também explica como trabalhar com GeneXus em diferentes cenários de integração: por um lado trazendo informações de um API de terceiros e por outro lado expondo com GeneXus o API que outros podem utilizar.

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