

Desenvolvimento de aplicativo para alerta de ocorrências de violência na cidade de Santa Maria-RS

Diego Luan Ferreira da Silva

Aluno do curso de Ciência da Computação - UFN
Santa Maria-RS Brasil
diegoluanfs@gmail.com

Herysson

Professor do curso de Ciência da Computação - UFN
Santa Maria-RS Brasil
herrysson@ufn.edu.br

Resumo—Este trabalho tem como objetivo propor o desenvolvimento de uma aplicação que seja possível facilitar o acesso aos treinos dos alunos dentro de uma academia, bem como, o aprimoramento da gestão de frequência e evolução de condicionamento físico dos usuários. Esse sistema tem como objetivo o desenvolvimento de uma ferramenta que possa ser disponibilizada ao público de forma online e para isso, foi utilizado de boas práticas do Scrum e do XP, bem como a ferramenta Astah de diagramação (UML) e o sistema de versionamento de código Github. Outro ponto importante é que a aplicação é uma Web API e conta com todo o desenvolvimento do back-end em C#.

I. INTRODUÇÃO

A demanda de pessoas por academias e espaços para se exercitarem, tem crescido de forma considerável nos últimos anos e isso tem impacto direto na economia que permeia esse nicho. Pois há uma busca pela melhoria da qualidade de vida e com o crescimento desse mercado, acaba crescendo o número de concorrentes e a competitividade também acaba aumentando. Assim, é necessário desenvolver técnicas para captar e reter os clientes e assim, motivá-los diariamente para reduzir os níveis de evasão dos alunos [1]. As formas mais comuns para realizar a captação de clientes em academias é pelo posicionamento da marca frente ao mercado, bem como a divulgação de serviços e o estreitamento na comunicação academia-cliente.

A. Objetivos

Esse trabalho tem como objetivo gerar uma ferramenta que permita o gestor da academia tornar mais intuitivo os exercícios para o aluno, na ausência de um profissional e maior segurança entre o aluno e os equipamentos.

Como objetivos específicos, esse trabalho tem os seguintes elementos a atingir no final do desenvolvimento:

- Desenvolver a diagramação das funcionalidades do sistema;
- Desenvolver no mínimo duas funcionalidades para realizar uma entrega para o gestor da academia;
- Gerar os scripts para a construção das tabelas do banco de dados;
- Avaliar a usabilidade da aplicação

II. REVISÃO BIBLIOGRÁFICA

Esta seção trata dos pilares fundamentais para a construção da aplicação que será desenvolvida, nessa etapa será apresentado brevemente os processos e também as tecnologias necessárias para a reprodução desse trabalho.

A. API e bibliotecas

É possível entender as API *Application Programming Interface* como uma padronização que permite a integração entre diferentes plataformas. A API serve como um contrato entre duas partes, sejam elas usuários ou servidores e nesse contrato são ditas as regras (normas e protocolos) que irão permitir a troca de informações entre os sistemas. Por exemplo, a API de geolocalização disponibilizado pela Google disponibiliza a padronização necessária, com normas e protocolos para a integração e utilização da mesma [?].

Normalmente, dentro dessas API's são disponibilizadas algumas bibliotecas, que são conjuntos de funções que facilitam o desenvolvimento.

B. Frameworks

Os *frameworks* são padronizações que facilitam no desenvolvimento e aceleram o andamento do projeto, visto que eles são conjuntos de funcionalidades genéricas que podem ser reutilizadas. São citadas como vantagens dos *frameworks* a facilidade na detecção de erros, otimização de recursos e reutilização de componentes classes, interfaces e métodos.

C. C#

O C# é uma linguagem, multiparadigma, fortemente tipada, desenvolvida pela Microsoft e é orientada a objetos, conta com o *framework* .NET, que auxilia no desenvolvimento dos projetos. O .NET é uma iniciativa da empresa Microsoft, que visa desenvolver uma plataforma única para o executar os sistemas e aplicações. Ou seja, todo código gerado para .NET pode ser executado em qualquer dispositivo que possua um *framework* da plataforma.

D. Github

Github é uma ferramenta de versionamento, disponibilizada pela Microsoft que auxilia no desenvolvimento, por se tratar de um serviço hospedado na nuvem que hospeda um sistema de versionamento de aplicações colaborativas e é multiplataforma. Com ele é possível obter um histórico de desenvolvimento, documentação da constante evolução do sistema e ainda permite que outros desenvolvedores colaborem com o projeto.

E. Bancos de dados

Atualmente o mercado apresenta diversos gerenciadores de banco de dados (SGBD), ele serve para coletar e manter os dados de forma segura. Nesse projeto vamos utilizar o SQL Server, visto que esse é mantido pela Microsoft, tendo uma versão gratuita para desenvolvimento e utilizações não comerciais.

F. Metodologias Ágeis

O desenvolvimento de *software* apresenta uma variedade de metodologias e técnicas que podem ser empregadas para a construção de aplicações. Essas metodologias servem para acelerar as entregas de um determinado projeto, elas fracionam a entrega do projeto para o cliente final em pequenos ciclos.

Como exemplo de metodologia ágil temos o Scrum, que divide o projeto em uma lista de funcionalidades (chamada de *Product Backlog*), com isso é possível montar as *Sprints Backlog*, que são conjuntos de funcionalidades que são agrupadas para serem entregues ao cliente ao final de uma *Sprint*, que normalmente dura 15 ou 21 dias. Ocorrem reuniões diariamente, conhecidas como *Daily* e podem haver incrementos de novas funcionalidades ao final de cada entrega.

Outra metodologia amplamente utilizada é a XP (*Extreme Programming*), é uma metodologia focada no desenvolvimento de *software* e pode ser utilizada de forma complementar ao Scrum, pois ela tem um conjunto de princípios, valores e práticas. São eles: comunicação, simplicidade, *feedback*, coragem e respeito.

G. Tecnologias

O desenvolvimento da interface, utiliza HTML5 (Linguagem de Marcação de Hipertexto), CSS (folhas de estilos em cascata), JavaScript, *frameworks* como Bootstrap e jQuery. A aplicação armazena e acessa os dados do SGBD SQLServer. Toda a informação trocada entre o *front-end* e o *back-end* será através de requisições HTTP, com a utilização de objetos no formato JSON. Assim, a aplicação contará o recurso de Minimal API apresentado no .NET 6 para consumir os valores informados ou solicitados pelo usuário.

III. METODOLOGIA

Será utilizado no desenvolvimento algumas boas práticas sustentadas pelas metodologias ágeis, SCRUM e XP. Como boas práticas do SCRUM utiliza-se a criação de *Backlog* contendo a lista de todas funcionalidades que o nosso projeto terá, listando e priorizando por nível de importância. Depois,

gera-se o *Sprint Backlog* para definir o tempo necessário para o desenvolvimento de cada funcionalidade. Como boas práticas do XP, é visto o design simples, para fazer um código simples e que já possa ser consumido nos testes iniciais, mesmo não sendo o produto final. Outra boa prática é a integração contínua, onde a cada término de desenvolvimento de uma nova funcionalidade incorporamos ao sistema e não esperamos o sistema ficar completamente concluído, pois isso permite conhecer o status real do projeto.

Os diagramas são utilizados para facilitar o desenvolvimento do projeto, entre eles temos o diagrama de caso de uso, ilustrado na Figura 2, que serve para descrever a funcionalidade proposta do projeto que será desenvolvido, para o levantamento dos requisitos funcionais e não funcionais. Para cada requisito funcional há um ou mais diagramas de sequência, sendo assim, geramos os diagramas para representar as funcionalidades, entre elas login, disposta na Figura ?? e

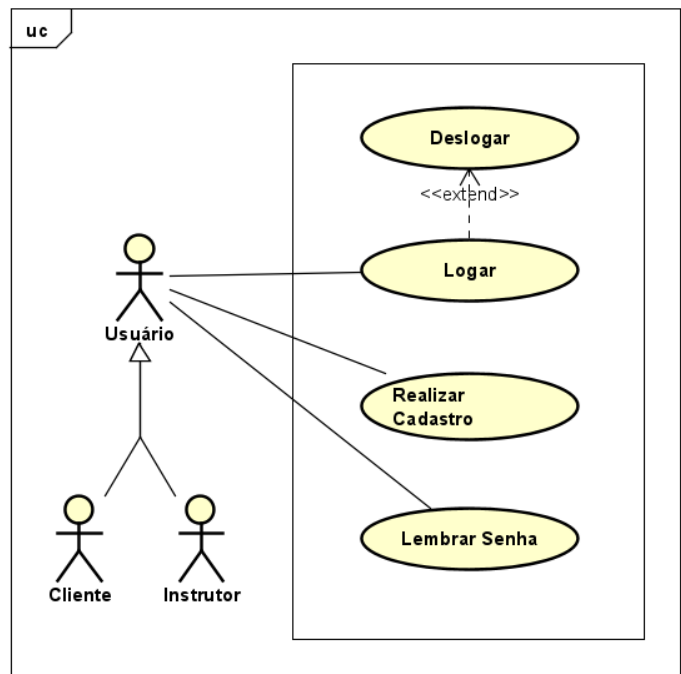


Figura 1. Diagrama de caso de uso

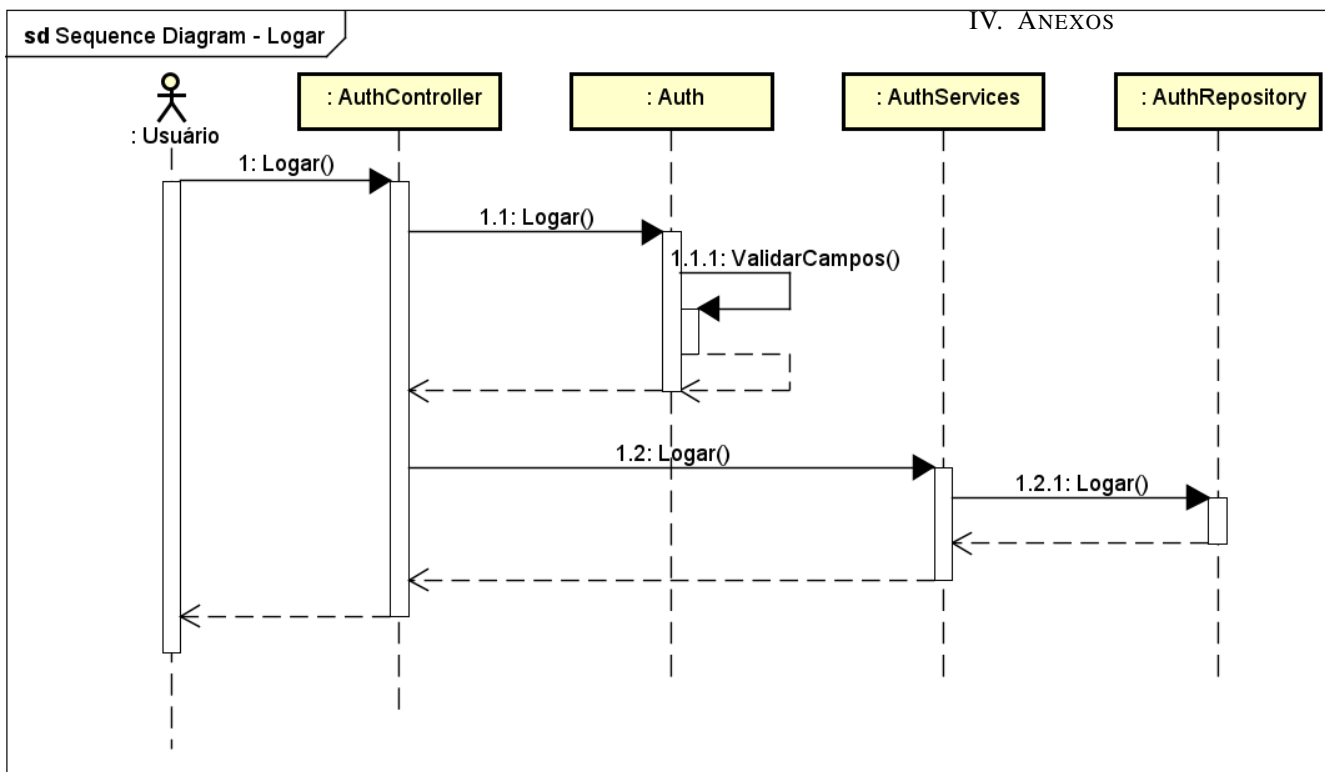


Figura 2. Diagrama de caso de uso

REFERÊNCIAS

- [1] R. de Azevedo Raiol, “Estratégias para aumentar captação, retenção e diminuir a evasão de clientes em academias.” *Revista Intercontinental de Gestão Desportiva*, vol. 10, no. 3, 2020.