# API – Back-End

## Roteiro de aula

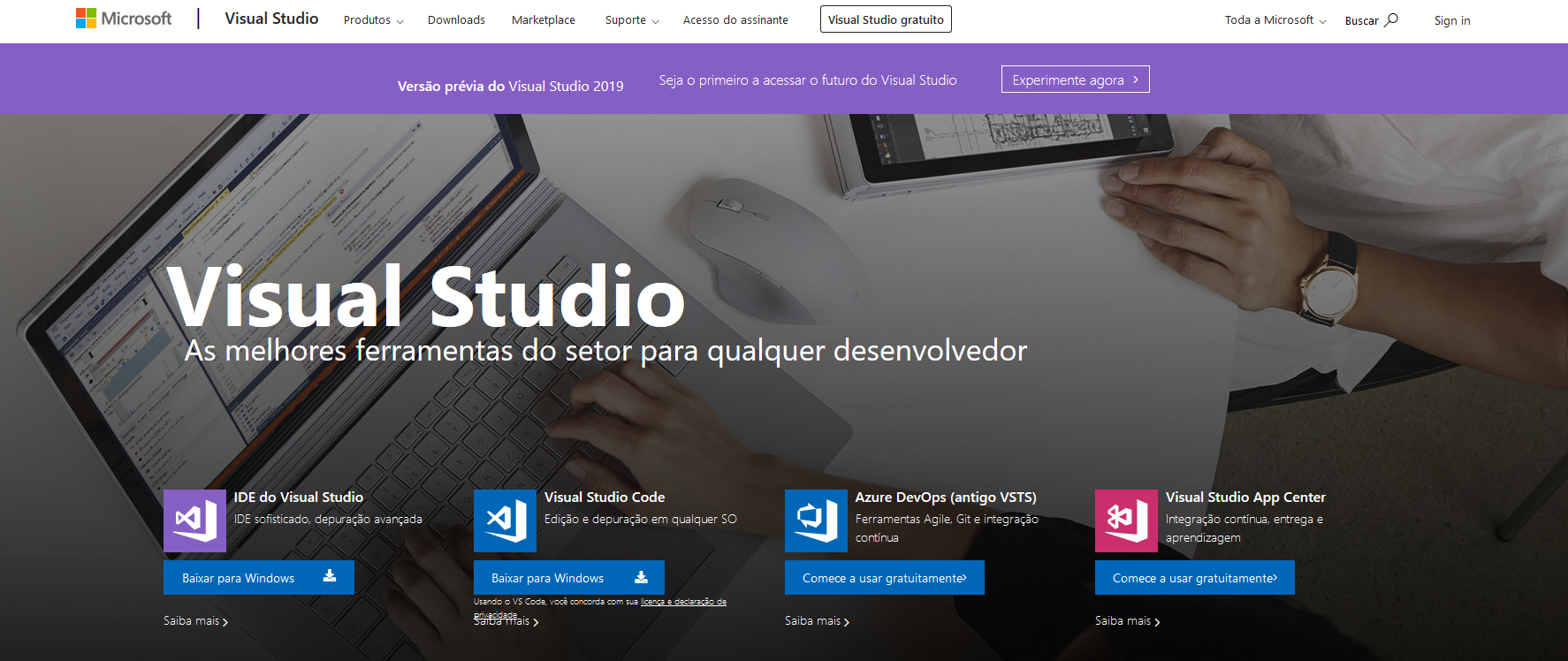
Obs.: Configuração do ambiente

Apresentação da ferramenta Visual Studio

<https://visualstudio.microsoft.com/pt-br/?rr=https%3A%2F%2Fwww.google.com%2F>

Apresentação do Azure DevOps – (antigo VSTS)

<https://azure.microsoft.com/pt-br/services/devops/>



Design patterns – boas práticas – DDD, TDD, BDD – fazer um desenho ilustrativo

Separação de responsabilidades

Microserviços

Configuração do projeto

Criar solução – Aplicativo Web .NET Core

Criar um projeto com um controller simples para apresentar o conceito de API/JSON

Mostrar o swagger no início do desenvolvimento

Criar o projeto de domínio

Criar todos os modelos

Criar o projeto de Repositório

Intercalando controller/repositório – sql server client

Usuário

JWT

Instituição

Tipos de Eventos

Eventos

Convites

Alterar para Entity Framework

Alterar o projeto para receber a imagem do usuário

-- desenvolvimento

Criar um novo projeto .NET Core – Aplicativo Web (API)

Vazio

Configurar startup.cs para a versão do .net a ser utilizada

Adicionar e colocar o mvc em uso

Adicionar pasta controllers

Adicionar um controlador vazio

Não esquecer do [Produces("application/json")]

Criar pasta de ViewModel

Adicionar um LoginViewModel

Logar com modelo

Incluir os outros verbos incluindo uma instância fixa

Incluir o Id posteriormente

Controller -> Domain (Modelo) -> Infra.Data (com os repositórios) , aqui entra o viewmodel?

Projeto MVC -> controller/viewmodel

Infra.data -> contexto/repositórios

Domain -> modelos das classes do banco

Só cria uma pasta ViewModels, Domain e referente a modelo do banco, viewmodel e o que recebe e envia da/para api. Fechou, isto mesmo.