



**UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”**

Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

JOÃO VITOR ROCHA SANTANA

**APLICATIVO PARA MONITORAMENTO E VISUALIZAÇÃO DE PROCESSOS NA
INDÚSTRIA 4.0**

Sorocaba

2022

JOÃO VITOR ROCHA SANTANA

**APLICATIVO PARA MONITORAMENTO E VISUALIZAÇÃO DE PROCESSOS
INDUSTRIAIS NA INDÚSTRIA 4.0**

Trabalho de Graduação apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Eduardo Paciência Godoy

Sorocaba

2022

S232a Santana, João Vitor Rocha

Aplicativo para monitoramento e visualização de processos na indústria 4.0 / João Vitor Rocha Santana. -- Sorocaba, 2022
62 f.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Eduardo Paciência Godoy

1. Aplicações industriais. 2. Realidade aumentada. 3. Aplicativos móveis. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

JOÃO VITOR ROCHA SANTANA

**APLICATIVO PARA MONITORAMENTO E VISUALIZAÇÃO DE PROCESSOS
INDUSTRIAIS NA INDÚSTRIA 4.0**

Trabalho de Graduação apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação”.

Comissão Examinadora

É fornecida pela Seção Técnica de Graduação e Pós-Graduação, Conselho de Curso ou Departamento
(folha será substituída)

Sorocaba
2022

AGRADECIMENTOS

Agradeço primeiramente a Deus por cumprir com a sua promessa na minha vida e por sempre ter me dado força ao longo da jornada.

Agradeço a toda minha família, em especial meus pais e minha irmã, pelos momentos de suporte, pela inspiração e por acreditarem em mim, sem vocês nada disso seria possível.

Gostaria de agradecer ao professor Eduardo Paciência Godoy por ter me dado a oportunidade de participar como aluno de iniciação científica e por ter confiado em mim para ser seu orientado, agradeço por todo suporte e ensinamentos.

Agradeço também a todos os professores do ICTS que passaram pela minha vida ao decorrer do curso e somaram no meu crescimento pessoal e profissional. Também deixo meu agradecimento a todos os técnicos e funcionários do ICTS que sempre estiveram dispostos a ajudar.

Por fim, mas não menos importante, agradeço a todos os alunos e amigos que dividiram essa jornada comigo, em especial a turma XVI e principalmente ao grupo "credibilidade" que sempre esteve comigo nos momentos bons e nos ruins.

SANTANA, J. V. R. **Aplicativo para Monitoramento e Visualização de Processos Industriais na Indústria 4.0**. 2022. 62 f. Trabalho de Graduação (Bacharel em Engenharia de Controle e Automação) - Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista, Sorocaba, 2022.

RESUMO

Atualmente o setor industrial passa pelo processo de quarta revolução industrial, ou Indústria 4.0 (I4.0), marcado pela utilização de tecnologias que visam tornar as fábricas mais inteligentes e conectadas. O principal objetivo da I4.0 é, através de sistemas que cooperam em tempo real, atingir uma melhor otimização de processos e assim garantir uma integração vertical, horizontal e de fim a fim da cadeia de valores. Para isso, entender o comportamento dos processos industriais para gerar eficiência na linha de produção e o fornecimento de insumos para tomadas de decisão na gestão de uma organização é fundamental. Dentre as diversas tecnologias habilitadoras da I4.0, a Realidade Aumentada (AR) vem se tornando uma alternativa cada vez mais utilizada. A AR sobrepõe as informações digitais ao mundo físico para fornecer uma melhor compreensão contextual de um ambiente para uma variedade de setores. Este trabalho objetiva o desenvolvimento de uma aplicação para monitoramento e visualização com auxílio de Realidade Aumentada de processos industriais para aplicações da I4.0. A solução consiste em um aplicativo para uso em dispositivos móveis que organiza o acesso a todas as informações pertinentes ao processo industrial, como documentação técnica do processo, *datasheets* e manuais de equipamentos, diagramas eletroeletrônicos e de instrumentação, juntamente com a possibilidade da utilização de AR para monitoramento do processo em tempo real. O desenvolvimento deste trabalho utilizou uma planta piloto de controle de processos industriais, o *framework* *Microservices* de microsserviços para a comunicação e aquisição dos dados do processo, a plataforma de desenvolvimento Unity para o desenvolvimento do aplicativo e do software *Vuforia* para o desenvolvimento da Realidade Aumentada. Testes operacionais do aplicativo em dispositivos móveis *Android* (smartphone e tablet) validaram o desenvolvimento onde notou-se uma facilidade ao acesso às informações e a visualização aprimorada da planta piloto com os dados em tempo real.

Palavras-chave: Indústria 4.0, Realidade Aumentada, Aplicativo, Monitoramento.

SANTANA, J. V. R. **Mobile Application for Monitoring and Visualization of Industrial Processes in Industry 4.0**. 2022. 62 f. Undergraduate final project (Bachelor in Control and Automation Engineering) – Institute of Science and Technology of Sorocaba, São Paulo State University, Sorocaba, 2022.

ABSTRACT

Currently, the industrial sector is undergoing a process of the fourth industrial revolution, or Industry 4.0 (I4.0), marked by the use of technologies aimed at making factories more intelligent and connected. The main objective of I4.0 is, through systems that cooperate in real time, to achieve better process optimization and thus guarantee vertical, horizontal and end-to-end integration of the value chain. For this, understanding the behavior of industrial processes to generate efficiency in the production line and the supply of inputs for decision-making in the management of an organization is fundamental. Among the various enabling technologies of I4.0, Augmented Reality has become an increasingly used alternative. AR overlays digital information on the physical world to provide a better contextual understanding of an environment for a variety of industries. This project aims to develop an application for monitoring and visualization with the aid of AR of industrial processes for I4.0 applications. The solution consists of a mobile application that organizes access to all information of the industrial process, such as technical documentation, datasheets and equipment manuals, electronic and instrumentation diagrams, along with the possibility of using AR to monitor the process in real time. The development of this project used an industrial process control pilot plant, the *Molecular* framework for microservices for the communication and data acquisition of process, the *Unity* development platform for the development of the mobile application and the *Vuforia* software for the development of the Augmented Reality. Operational tests of the application on Android mobile devices (smartphone and tablet) validated the development where *easy* access to information and improved visualization of the pilot plant with real-time data was noted.

Keywords: Industry 4.0, Augmented Reality, Application, Monitoring.

LISTA DE FIGURAS

FIGURA 1 - Mudança de Arquitetura de Integração Industrial.....	14
FIGURA 2 - Exemplo de Aplicação de Realidade Aumentada.	16
FIGURA 3 - Representação simplificada de Continuum Realidade-Virtualidade.....	19
FIGURA 4 - Interação entre usuário e dispositivo AR.	20
FIGURA 5 - Esquemático da solução proposta.....	21
FIGURA 6 - Bancada de processos: Imagem e Diagrama PI&D.....	22
FIGURA 7 - Esquemático da solução.	27
FIGURA 8 - Tela do menu inicial.	29
FIGURA 9 - Script do menu inicial.....	30
FIGURA 10 - Função do botão no menu inicial.	30
FIGURA 11 - Fluxo do desenvolvimento.	31
FIGURA 12 - Geração do <i>Model Target</i>	32
FIGURA 13 - Fluxo visual do aplicativo.	33
FIGURA 14 - Importação do MTG no <i>Unity</i>	33
FIGURA 15 - Início do desenvolvimento visual no <i>Unity</i>	34
FIGURA 16 - Desenvolvimento visual no <i>Unity</i>	35
FIGURA 17 - Desenvolvimento visual por sensor no <i>Unity</i>	35
FIGURA 18 - Primeira parte da comunicação.	36
FIGURA 19 - Classe de configuração da comunicação.	37
FIGURA 20 - Função <i>SendRequest</i>	37
FIGURA 21 - Co-rotina criada.	38
FIGURA 22 - Finalização da comunicação.....	39
FIGURA 23 - Métodos para requisição de entradas e saídas.	39
FIGURA 24 - Fluxo da segunda parte da comunicação.	40
FIGURA 25 - Script para requisição em tempo real	41
FIGURA 26 - Função <i>CallAPI</i>	41
FIGURA 27 - Função <i>ShowMessage</i>	42
FIGURA 28 - Configuração do scrip e label.	43
FIGURA 29 - Fluxo de funcionamento da abertura dos datasheets.	44
FIGURA 30 - Scrip datasheets	44
FIGURA 31 - Configuração botão do datasheets.	45
FIGURA 32 - Tela padrão das funcionalidades.	46

FIGURA 33 - Fluxo de funcionamento das funcionalidades.	46
FIGURA 34 - UniWebView para abertura dos documentos.	47
FIGURA 35 - UniWebView para abertura do dashboard.	48
FIGURA 36 - UniWebView para abertura do PLN.	48
FIGURA 37 - Configuração do APK.	49
FIGURA 38 - Gerando o APK.	50
FIGURA 39 - Menu Inicial.	51
FIGURA 40 - Dashboard.	52
FIGURA 41 - Realidade Aumentada.	53
FIGURA 42 - Visualização dos dados na Realidade Aumentada.	54
FIGURA 43 - Documentos.	55
FIGURA 44 - Consulta de dados.	56

LISTA DE TABELAS

TABELA 1- Endereços requisitados pelo serviço <i>Influx</i>	36
---	----

LISTA DE ABREVIATURAS E SIGLAS

CNI	Confederação Nacional da Indústria
PIB	Produto Interno Bruto
IBGE	Instituto Brasileiro de Geografia e Estatística
AR	Realidade Aumentada
I4.0	Indústria 4.0
CPS	Sistemas Ciber-Físicos
BD	Big Data
IoT	Internet das Coisas
IoS	Internet como serviço
CS	Cibersegurança
CC	Computação em Nuvem
XR	Realidade Estendida
VR	Realidade Virtual
MR	Realidade Mista
API	Interfaces de Programação de Aplicativos
IDE	Ambiente de desenvolvimento integrado
TMP	TextMeshPro
SDK	Software Development Kit
MTG	Model Target Generator
VB	Visual Basic
IHM	Integração Homem Máquina
PLN	Processamento de Língua Natural
JDK	Java Development Kit
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos.....	17
1.2	Estrutura do Trabalho.....	17
2	REVISÃO DE LITERATURA.....	18
2.1	Indústria 4.0.....	18
2.2	Realidade Estendida.....	18
3	MATERIAIS E TECNOLOGIAS	21
3.1	Proposta do Trabalho	21
3.2	Planta Piloto 4.0.....	22
3.2.1	<i>Framework Molecular</i>.....	23
3.3	Ferramentas de Software.....	24
3.3.1	<i>Unity</i>.....	24
3.3.2	<i>Vuforia</i>	25
3.3.3	<i>Virtual Studio</i>	26
4	DESENVOLVIMENTO	27
4.1	Metodologia.....	27
4.2	Menu Inicial do Aplicativo	28
4.3	Realidade Aumentada.....	30
4.3.1	<i>Target</i>	31
4.3.2	Desenvolvimento Visual	32
4.3.3	Comunicação com a API.....	35
4.3.3.1	Requisição dos Dados de Variáveis do Processo	36
4.3.3.2	Comunicação em Tempo Real	40
4.3.4	Datasheets.....	43
4.4	Funcionalidades	45
4.4.1	Datasheets e Documentação	46
4.4.2	Dashboard	47
4.4.3	Consulta de dados.....	48
4.5	Geração do Aplicativo <i>Android</i> (APK)	49
5	RESULTADOS E DISCUSSÕES	51
5.1	Aplicativo	51

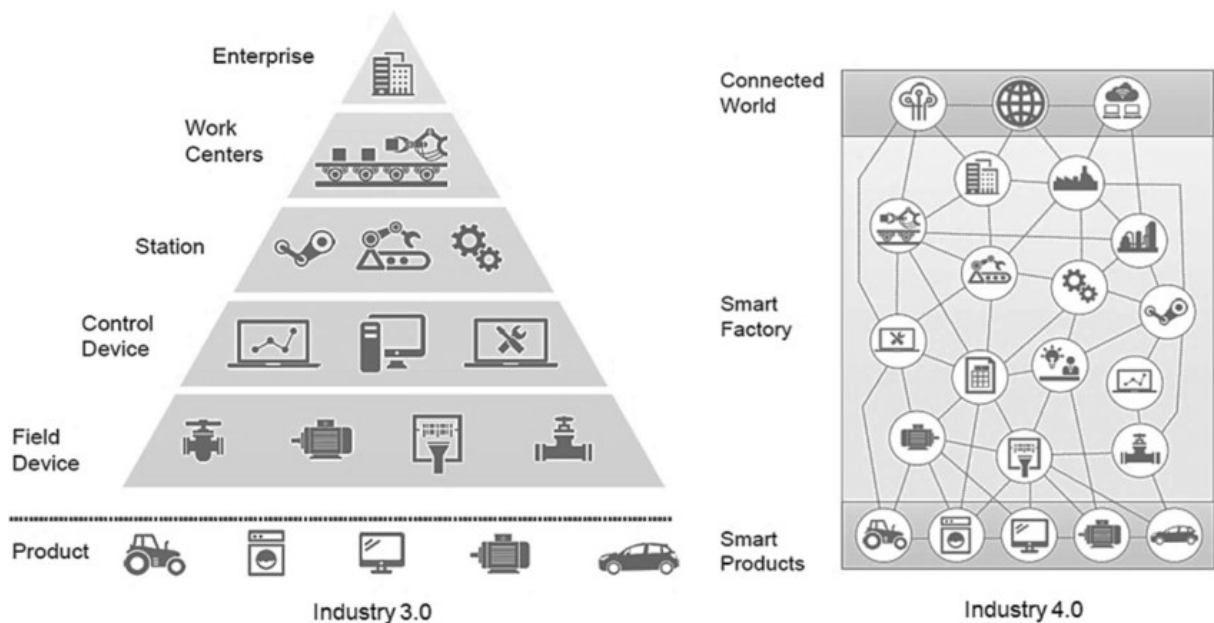
5.1.1	Menu Inicial	51
5.1.2	Dashboard.....	52
5.1.3	Realidade Aumentada.....	53
5.1.4	Documentação	54
5.1.5	Assistente Virtual para Consulta de Dados	55
5.2	Considerações Finais	56
6	CONCLUSÃO	58
	REFERÊNCIAS.....	59

1 INTRODUÇÃO

Ao longo dos anos, diversos avanços tecnológicos permitiram grandes mudanças no cenário industrial, sendo estas marcadas por movimentos denominados revoluções industriais. A terceira revolução industrial foi marcada pela introdução da automação através da utilização de robôs, sensores e controladores, contudo, apesar do grande desenvolvimento alcançado, os sistemas ainda não apresentavam uma conectividade entre si.

A partir desta deficiência surge a quarta revolução industrial, em que através do desenvolvimento de tecnologias habilitadoras objetiva a implementação da conectividade entre diversos processos existentes e, desta forma, os torna digitalmente interligados através de sistemas ciberfísicos, permitindo o monitoramento online dos mesmos. Através da Figura 1 é possível notar como a quarta revolução industrial interferiu diretamente na comunicação dos processos, quebrando a hierarquia tradicional da Indústria 3.0.

Figura 1 - Mudança de Arquitetura de Integração Industrial.



Fonte: Frère et al, 2018.

A quarta revolução industrial também pode ser chamada de Indústria 4.0, termo este de origem alemã que foi utilizado pela primeira vez em 2011 durante a Feira de Hannover, principal evento mundial de tecnologias para a indústria, como forma de sintetizar as mudanças promovidas nos processos industriais através da informatização (ANPEI, 2019). Apesar de não ser uma ideia nova, o desenvolvimento da Indústria 4.0 no Brasil enfrenta diversos desafios que

vão desde os investimentos em equipamentos, à mudanças de layouts, adaptação de processos, criação de novas especialidades, desenvolvimento de competências, entre outras (PORTAL DA INDÚSTRIA, 2022).

De acordo com a Confederação Nacional da Indústria (CNI), o setor industrial representa 21,4% do PIB do Brasil e responde por 69,2% das exportações de bens e serviços do país, contudo, através do estudo realizado pela CNI, dos 24 setores industriais do Brasil, 14 estão atrasados na adoção de tecnologias, sendo que os mesmos são responsáveis por cerca de 40% de toda a produção industrial, segundo o IBGE (V2COM, 2021). Desta forma, fica nítido como a adoção das tecnologias da Indústria 4.0 implicam diretamente no PIB brasileiro, de acordo com a Accenture, a implementação das tecnologias ligadas à Internet das Coisas deverá impactar em aproximadamente US\$ 39 bilhões no PIB até 2030 (PORTAL DA INDÚSTRIA, 2022).

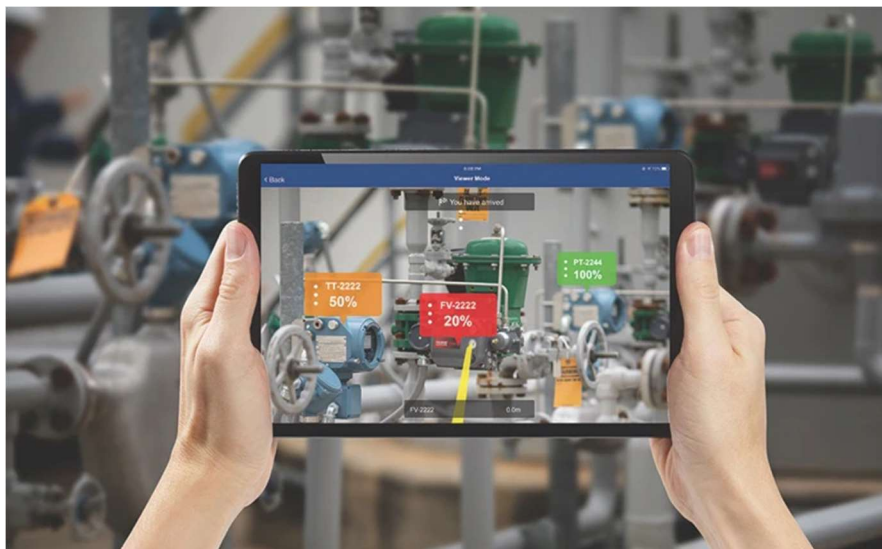
A implementação da quarta revolução industrial está diretamente ligada ao monitoramento contínuo de processos, de acordo com a consultoria *McKinsey*, até 2025, os processos relacionados à Indústria 4.0 poderão reduzir custos de manutenção de equipamentos entre 10% e 40%, reduzir o consumo de energia entre 10% e 20% e aumentar a eficiência do trabalho entre 10% e 25% (PORTAL DA INDÚSTRIA, 2022). Desta maneira, nota-se que através da implementação de um monitoramento rápido, fácil e eficaz, é possível gerar um grande aumento na produtividade da empresa e, conseqüentemente, uma redução dos custos de processo.

Dentre as diversas tecnologias utilizadas para o monitoramento, a Realidade Aumentada (AR – *Augmented Reality*) vem se tornando uma alternativa cada vez mais utilizada. A AR sobrepõe as informações digitais ao mundo físico para fornecer uma melhor compreensão contextual de um ambiente para uma variedade de setores (TOSHNIWAL; DASTIDAR, 2014). A AR pode ser utilizada em diversos setores dentro da indústria, sendo sua forma mais comum para a visualização e monitoramento de processos através da junção dos dados virtuais com o processo em si. A Figura 2 exemplifica a utilização da Realidade Aumentada no ambiente industrial.

A utilização dos dados proporciona uma melhor visualização e compreensão do sistema ou objeto necessário (SREEKANTA et al, 2020). Além disso, a sobreposição dos dados ao mundo real ajuda na melhora da interação do usuário com o sistema permitindo que o mesmo realize inspeção, manutenção e facilite a colaboração/cooperação com máquinas remotas e especialistas (RUNJI; LIN, 2020). Nesse sentido, a comunicação móvel mostra-se uma alternativa cada vez mais explorada dentro do contexto da Indústria 4.0, pois os aplicativos moveis permitem o acesso e a modificação de dados corporativos a qualquer momento. No entanto, apesar

de facilitar essa interação, um desafio a ser superado se encontra na conectividade necessária entre essas aplicações móveis e os sistemas e equipamentos utilizados nas indústrias.

Figura 2 - Exemplo de Aplicação de Realidade Aumentada.



Fonte: (AUTOMATION WORD, 2021)

É notório que a utilização das tecnologias habilitadoras da Indústria 4.0 gera um grande avanço em produtividade e competitividade para a indústria, contudo, o cenário atual do país não favorece a utilização das mesmas. Sendo assim, este trabalho tem o intuito de auxiliar o desenvolvimento dessas tecnologias no Brasil, alavancar a transição para a Indústria 4.0 e apresentar-se como um norte para novos trabalhos e pesquisas pertinentes desse tema. Além disso, visa desenvolver uma alternativa de monitoramento de baixo custo e alta aplicabilidade pois tratando-se de um aplicativo enfrenta uma barreira de entrada menor.

Desta forma, busca-se desenvolver um aplicativo mobile visando possibilitar o monitoramento e o acesso organizado às informações de uma planta piloto de controle de processos industriais aderente à Indústria 4.0, além de implementar sua visualização através de Realidade Aumentada. A conectividade entre o aplicativo e as informações do processo industrial é realizada através da comunicação via serviços. Espera-se assim desenvolver uma alternativa de baixo custo, através da utilização de softwares gratuitos, e fácil acesso para fomentar o avanço tecnológico e acelerar o processo de implementação da Indústria 4.0 no Brasil.

1.1 Objetivos

O presente trabalho tem como objetivo o desenvolvimento de um aplicativo *Android* para uso em dispositivos móveis com a finalidade de visualização, monitoramento e organização do acesso às informações de processos industriais para aplicações da Indústria 4.0. O aplicativo utiliza Realidade Aumentada para visualização aprimorada e monitoramento em tempo real de dados operacionais e variáveis de processo. Adicionalmente, o aplicativo permite e organiza o acesso a diversas informações e documentação técnica do processo, como datasheets e manuais de equipamentos, diagramas eletroeletrônicos e diagramas de processos e instrumentação (P&ID).

1.2 Estrutura do Trabalho

O presente trabalho contém cinco capítulos além do resumo. O primeiro capítulo traz uma introdução do tema abordado com a apresentação dos objetivos pretendidos e uma visão de como está documentada a estrutura do trabalho. O segundo capítulo traz uma revisão da literatura, sendo abordados vários conceitos e revisado o que literatura possui sobre o tema, também se discute a motivação para o uso de algumas tecnologias.

No Capítulo 3 é apresentado a proposta do projeto, descrevendo a ideia desenvolvida e apresentado os softwares e tecnologias. Nesta seção também está contida a apresentação da bancada de processos utilizada.

No Capítulo 4, após apresentado os conceitos e materiais a serem utilizados, é documentado todo o desenvolvimento da proposta. Iniciando-se pela estruturação da solução juntamente com a metodologia aplicada no desenvolvimento, passando pelo desenvolvimento das funcionalidades contidas e a criação de toda a realidade aumentada, além da comunicação necessária para o monitoramento do processo em tempo real.

O quinto capítulo traz os resultados obtidos juntamente com as respectivas descrições e exibições do mesmo rodando em um smartphone *Android*. O último capítulo apresenta as conclusões deste trabalho contendo o objetivo atingido com os resultados esperados. Por fim, estão as referências bibliográficas utilizadas para embasamento e suporte deste trabalho.

2 REVISÃO DE LITERATURA

Para início do desenvolvimento do trabalho, a compreensão do tema abordado e conhecimento de distintos trabalhos do campo é indispensável, fazendo-se necessário o levantamento de outras produções acadêmicas dentro do tema proposto.

2.1 Indústria 4.0

O termo revolução industrial representa um conceito que muda essencialmente a nossa sociedade e economia (BLOEM et al, 2014). A quarta revolução industrial foi apresentada pelo governo alemão em 2011 com uma nova rubrica denominada Indústria 4.0 (I4.0) (WAGNER et al, 2017; LU, 2017; GRIECO et al, 2017; MOTYL et al, 2017; WEYER et al, 2015).

A I4.0 é caracterizada em manufatura e serviços por processos altamente desenvolvidos de automação, digitalização, eletrônica e TI (LU, 2017). Do ponto de vista da produção e gerenciamento de serviços, a I4.0 foca em estabelecer sistemas inteligentes e comunicativos como *Machine-to-Machine* e interação Homem-Máquina, lidando com o fluxo de dados da interação de sistemas inteligentes e distribuídos (SALKIN et al, 2018).

O objetivo da I4.0 é trabalhar com um maior nível de automatização, alcançando um maior nível de produtividade e eficiência operacional (LU, 2017; PERUZZINI et al, 2017), conectando o mundo físico ao virtual (LEYH et al, 2017; BAENA et al, 2017).

Segundo vários autores (LU, 2017; MOTYL et al, 2017; WEYER et al, 2015), o termo I4.0 pode ser assumido como produção de Sistemas Ciber-Físicos (CPS), baseado em dados heterogêneos e integração de conhecimento e pode ser resumido como um processo de fabricação interoperável, integrado, adaptado, otimizado, orientado a serviços que está correlacionada com algoritmos, Big Data (BD) e altas tecnologias como Internet das Coisas (IoT) e Serviços (IoS), Automação Industrial, Cibersegurança (CS), Computação em Nuvem (CC) ou Robótica Inteligente (LU, 2017; PERUZZINI et al, 2017; BAENA et al, 2017).

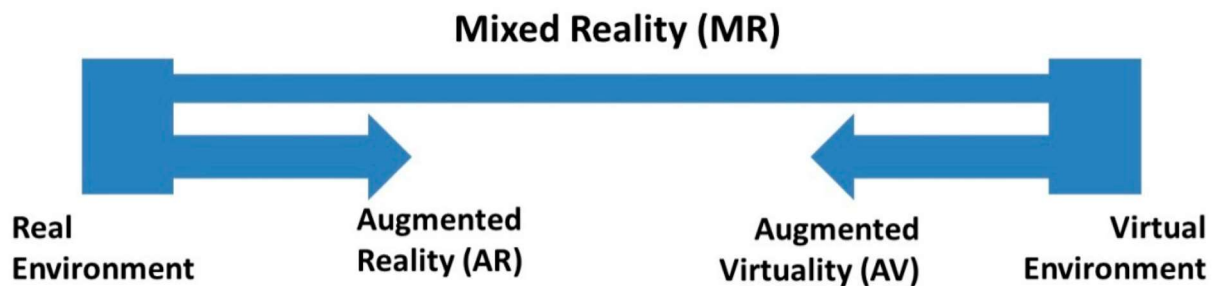
2.2 Realidade Estendida

A realidade estendida (XR), também conhecida como “realidade cruzada” (SHERMAN; CARIG, 2019), é um termo utilizado para abranger diversos tipos de tecnologias como a realidade virtual (VR), aumentada (AR) e mista (MR) (FAST-BERGLUND et al. 2018), e tornou-se mais comum durante as três últimas décadas.

A VR implica em uma tecnologia onde o usuário está totalmente imerso em um mundo sintético, produzido digitalmente, perdendo a referência sobre o mundo real. As primeiras conceituações e ideias sobre VR foram desenvolvidas já na década de 1960 (AUKSTAKALNISs, 2017; BERG; VANCE, 2017; WHYTE; NIKOLĆ, 2018) e o desenvolvimento de aplicativos começou na década de 1990 como várias soluções tecnológicas avançadas (MOORE, 1995a; 1998b).

A definição de Realidade Mista trata da relação entre o real e o virtual. Considerando-se uma analogia em que o ambiente real e o ambiente virtual são extremos, a ponte que conecta o real e o virtual é preenchida por tecnologias de MR que são capazes de mesclar conteúdo virtual ao mundo real ou vice-versa. Esta analogia é representada pelo trabalho seminal de Paul Milgram (1995) descrevendo o “Continuum Realidade-Virtualidade” e pode ser observada na Figura 3.

Figura 3 - Representação simplificada de Continuum Realidade-Virtualidade.



Fonte: Milgram et al. 1995.

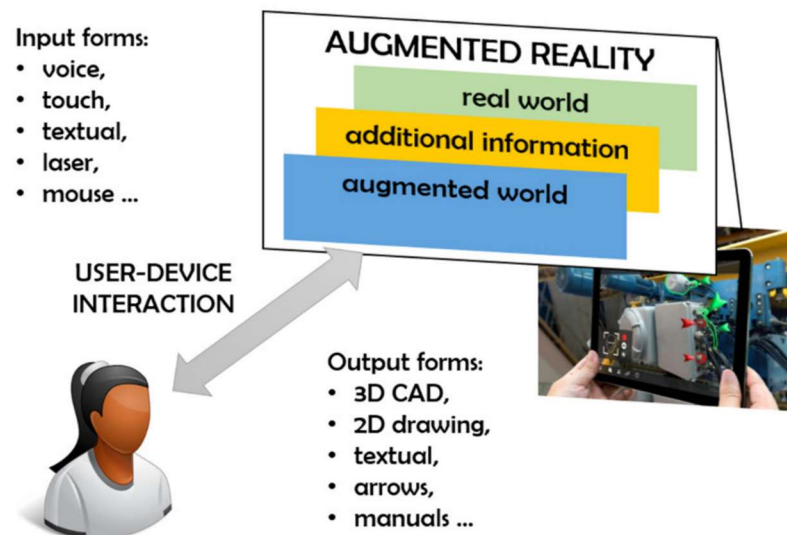
A AR é uma tecnologia em rápida evolução para a criação de uma realidade digital na qual o ambiente físico do usuário é complementado com conteúdo digital adicional, como mensagens de texto, imagens, objetos virtuais 3D, vídeos, etc., no campo de visão do usuário (SYBERFELDT et al, 2016; MASOOD; EGGER, 2020; CARMIGNIANI et al, 2011). Sherman e Craig (2019) incluíram a AR como parte da VR, no entanto, alguns dos estudos recentes veem AR e VR como tecnologias “irmãs” separadas (WANG et al., 2018) e, de acordo com Azuma (1997, p.2), a diferença crucial entre VR e AR é que “AR suplementa a realidade, em vez de substituí-la completamente”.

O princípio da AR é a combinação de dois cenários: 1) realidade processada digitalmente com 2) objetos artificiais adicionados digitalmente que podem ser objetos planos 2D, ou por outras definições que consideram apenas objetos 3D dentro da cena (HOŘEJŠÍ, 2015). Além disso, pode definir-se como as características do sistema AR a capacidade de combinar objetos reais e virtuais em um ambiente real, a capacidade de alinhar entre si os objetos reais e virtuais

e a capacidade em execução interativa, em 3D e em tempo real (SYBERFELDT et al, 2015; SYBERFELDT et al; 2016).

Uma aplicação típica de AR consiste em três camadas, conforme mostrado na Figura 4. A camada inferior representa o mundo real e material. Na camada intermediária, informações digitais relevantes e úteis são adicionadas ao mundo real. A combinação dessas duas camadas cria um terceiro mundo aumentado no qual os usuários têm acesso ao conteúdo digital. As aplicações de AR combinam objetos reais e virtuais e fornecem interação em tempo real entre os usuários e o mundo aumentado e o alinhamento geométrico de objetos virtuais e reais (DINI; DALLE MURA, 2018).

Figura 4 - Interação entre usuário e dispositivo AR.



Fonte: Bajči et al, 2018 .

As partes essenciais de um sistema de AR fazem uso de dispositivos eletrônicos para visualizar direta ou indiretamente uma combinação do mundo real com elementos virtuais. Segundo Fraga-Lamas (2018), esses elementos podem ser:

- Elemento de captura de imagem – câmera web é suficiente (HOŘEJŠÍ, 2015);
- Display – para projeção das informações virtuais sobre as imagens adquiridas pelo elemento de captura de imagem.(SYBERFELDT et al, 2016; SYBERFELDT et al; 2017);
- Unidade de processamento – para gerar informações virtuais a serem projetadas;
- Elementos de ativação – para acionar a exibição de informações virtuais, por exemplo, sensores, marcadores QR, posições GPS, imagens, etc.

3 MATERIAIS E TECNOLOGIAS

Nesta subseção são apresentadas as descrições da proposta do trabalho, dos materiais, tecnologias utilizadas e configurações de ambientes para desenvolvimento.

3.1 Proposta do Trabalho

O presente trabalho visa desenvolver uma alternativa de baixo custo e alta aplicabilidade para monitoramento de dados e visualização através de realidade aumentada de processos industriais. Para isso, utilizou-se como processo industrial, uma planta piloto de controle de processos industriais presente no laboratório do Grupo de Automação e Sistemas Integráveis (GASI) na Unesp Sorocaba.

A planta piloto de processos industrial possui uma arquitetura orientada a microsserviços os quais são responsáveis pela aquisição de dados oriundos dos dispositivos que nela estão instalados, desta forma, para o monitoramento propõe-se a criação de um aplicativo que reúne todas as informações e dados pertinentes à bancada, sendo estes toda a documentação dos sensores e atuadores, a consulta de dados históricos e também a visualização dos dados medidos em tempo real, de modo a facilitar e centralizar os mesmos.

Uma outra alternativa proposta para o monitoramento é a utilização de Realidade Aumentada para visualização dos dados em tempo real, possibilitando assim uma alternativa que traz uma melhor interpretação dos dados, além da exibição junto ao respectivo sensor e a possibilidade de abertura dos datasheets. A Figura 5 exemplifica o esquemático da proposta do trabalho.

Figura 5 - Esquemático da solução proposta.



Fonte: Autoria própria.

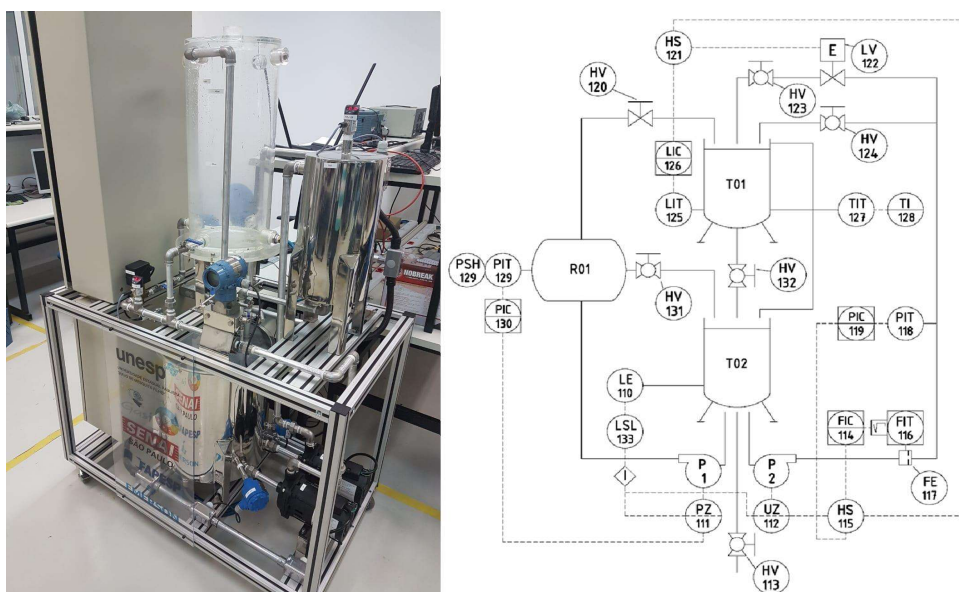
Para que fosse possível atingir os resultados propostos utilizou-se de ferramentas de software que possibilitem o desenvolvimento da realidade aumentada, como por exemplo o *Unity* e o *Vuforia*, além de serviços do *framework Molecular* para a requisição dos dados através de uma API padronizada .

3.2 Planta Piloto 4.0

A bancada de processos usada foi construída no projeto FAPESP 2008/19984-4, do qual este projeto de PFC está atrelado. Essa bancada, representada na Figura 6, possui diversas malhas de controle (pressão, temperatura, nível e vazão) e foi desenvolvida para uso do *framework Molecular* como arquitetura de automação e controle para aplicações de Indústria 4.0 e Internet das Coisas Industrial (BIGHETI et al., 2018). Pode-se observar uma vista da planta piloto, onde é possível ver todos os seus componentes como atuadores e sensores, além de seu PI&D (*Piping and Instrumentation Diagram*) para uma melhor visualização de suas malhas de controle.

A parte física da planta piloto possui um tanque em inox principal (TQ02) de 83 litros, instalado na parte inferior da planta para armazenamento de água. O líquido pode ser bombeado através de tubos de inox de ½” para parte superior da planta em duas rotas distintas, tanto para o tanque em acrílico (TQ01) de 38 litros com a bomba (P2), quanto para o reservatório de pressão em inox (R01) de 15 litros com a bomba (P1).

Figura 6 - Bancada de processos: Imagem e Diagrama PI&D.



Fonte: Acervo pessoal.

As variáveis de processos na planta e seus respectivos instrumentos de medição da Emerson são: nível do tanque (LIT125) e vazão (FIT116) com o transmissor de pressão coplanar de 0 a 5000 mmH₂O da ROSEMOUNT modelo 2051CD, pressão de linha (PIT118) e pressão do reservatório (PIT129) com o pressostato eletrônico de 0 a 2,5 bar da WIKA modelo PSD-4, e temperatura (TIT127). Para medição de vazão é usada uma placa de orifício junto ao transmissor de pressão.

Como variáveis de comando, tem-se duas bombas e uma válvula proporcional. As bombas de baixo ruído Dancor modelo Pratika CP-4R (P1 e P2) têm as mesmas características de potência de 0,5CV e uma vazão de 4,24 m³/h, sendo controladas por inversores de frequência da WEG modelo CFW300. Para evitar que as bombas funcionem sem água no reservatório adicionou-se uma chave de nível (LSL110) da SITRON. A válvula proporcional motorizada (LV 122) de ½” é da BUSCHJOST modelo 8288200.9650.

3.2.1 *Framework Molecular*

Para a comunicação e coleta de dados da bancada de processos utilizou-se do *framework Molecular*. O *Molecular* é uma estrutura de desenvolvimento com a linguagem *JavaScript* para desenvolvimento de aplicações orientadas a micro serviços. O *framework* é de código aberto e roda sobre a plataforma Node.js. A comunicação entre os microserviços é realizada via um serviço de mensagens (*Transporter*) também conhecido como Broker de mensagens. O serviço de mensagens (*Transporter*) é um dos componentes mais importantes da arquitetura e da infraestrutura para a mensageria, pois é o mecanismo que coordena o envio e a recepção de mensagens em uma fila (MOLEUCULER, 2022).

O *Transporter* é capaz de enfileirar e manter as mensagens até serem processadas por consumidores. O produtor simplesmente envia a mensagem para o *Transporter*, sem a necessidade de um mecanismo de descoberta de serviços. *API Gateway* é um microserviço de requisição de dados remota, que padroniza uma API via padrão REST para requisição de dados remotos de serviços desenvolvidos na plataforma *Molecular*. Através do *framework Molecular* e de requerimentos web foi possível obter as informações em tempo real da bancada de processos.

3.3 Ferramentas de Software

São utilizados *softwares* que fornecem a estrutura necessária para o desenvolvimento do trabalho, por exemplo, o sistema operacional *Windows 10*. Todos os *softwares* apresentados estão disponíveis na *internet*.

3.3.1 *Unity*

Para o desenvolvimento do aplicativo, devido a aplicação em realidade aumentada, optou-se pela utilização da *engine* da *Unity* visto que a mesma se mostrou a mais efetiva na comunicação com o *software* de desenvolvimento de AR (*Vuforia*), além de possuir uma ampla base de projetos já desenvolvidos que serviram como base de estudos para o presente projeto. O *Unity* é uma *game engine* multiplataforma para jogos 2D e 3D, criado pela *Unity Technologies* em 2005 para “democratizar” a indústria de desenvolvimento de jogos, tentando torná-lo acessível a todos os desenvolvedores (INTERSOG, 2020).

Basicamente, *Unity* é um conjunto de ferramentas de *software*, Interfaces de Programação de Aplicativos (APIs) e pacotes que são projetados para criar jogos 2D ou 3D para diferentes plataformas, como por exemplo *Windows*, *Linux*, *Android* e *iOS*. O *Unity* fornece um ambiente de desenvolvimento integrado (IDE) agradável no qual você pode criar seu jogo rapidamente. Devido às suas características, o *Unity* foi feito para ser gratuito e fácil de instalar, este motor de jogo vem agitando o mundo do desenvolvimento de jogos desde o seu lançamento (LINKEDIN, 2019).

Apesar do foco em desenvolvimento de jogos, com o passar do tempo, o *Unity* foi evoluindo a sua IDE e hoje conta com aplicações em diversas áreas, sendo a realidade aumentada uma delas. O *Unity* usa C# como linguagem de programação e utiliza da sua fácil integração com o *Visual Studio* para escrever e depurar os códigos (INTERSOG, 2020).

UniWebView

O *UniWebView* é um plug-in para a *Unity* que tem como funcionalidade principal adicionar um componente de visualização da web aos projetos. Ele fornece um conjunto de APIs abstratas, multiplataformas e de alto nível em C# (UNIWEBVIEW, 2022). Os objetos *WebView* permitem exibir conteúdo da Web como parte do layout do aplicativo e apresenta-se como uma alternativa útil para ter maior controle sobre a interface do usuário, além de possuir opções de

configuração avançadas que permitem incorporar páginas da Web em um ambiente especialmente projetado para o aplicativo (DEVELOPERS ANDROID, 2022).

TextMeshPro

O TextMeshPro (também conhecido como TMP) é uma solução de texto definitiva para *Unity*. O TMP utiliza um campo de distância assinado para renderizar o texto na tela, desta forma o texto aparece nítido, independentemente de seu tamanho relativo e distância da câmera (RAYWENDERLICH, 2021). Juntamente com um conjunto de *shaders* personalizados, o TMP oferece melhorias substanciais na qualidade visual e uma flexibilidade maior quando se trata de estilo e textura de texto (UNITY TECHNOLOGIES, 2022).

3.3.2 Vuforia

Para o desenvolvimento da realidade aumentada, juntamente com o *Unity*, utilizou-se do *Vuforia*. O *Vuforia* é um kit de desenvolvimento de *software* de realidade aumentada (SDK) para dispositivos móveis que permite a criação de aplicativos de realidade aumentada e foi adquirido pela PTC Inc. em novembro de 2015 (PTC, 2022). O *Vuforia* com *Unity* fornece um SDK de desenvolvimento AR multiplataforma com suporte para a maioria dos dispositivos *Android* e *iOS* existentes. Pela grande quantidade de dispositivos suportados, o *Vuforia* permite adicionar funcionalidades de visão computacional que vão além dos seus concorrentes (TUTORIALS FOR AR, 2019).

Para o reconhecimento e interpretação de imagens, o *Vuforia* utiliza tecnologia de visão computacional para reconhecer e rastrear imagens planares e objetos 3D em tempo real. Esse recurso de registro de imagem permite que os desenvolvedores posicionem e orientem objetos virtuais, como modelos 3D e outras mídias, em relação a objetos do mundo real quando vistos pela câmera de um dispositivo móvel. O objeto virtual rastreia a posição e orientação da imagem em tempo real para que a perspectiva do observador no objeto corresponda à perspectiva do alvo. Assim, parece que o objeto virtual é uma parte da cena do mundo real (PROGRAMMING HISTORIAN. 2021).

Model Target Generator

O *Model Target Generator* (MTG) converte um modelo 3D existente em um banco de dados do *Vuforia Engine* que pode ser usado para rastreamento do mesmo. Essa ferramenta

permite confirmar se os recursos do seu modelo serão utilizados, configurar a posição de encaixe inicial e exportar o banco de dados final. O banco de dados gerado pelo MTG inclui renderização do modelo do *Guide View* (posição de detecção), além de ser treinado com o MTG por meio de um processo de aprendizado profundo baseado em nuvem que permite o aplicativo alternar automaticamente entre vários alvos de modelo (VUFORIA; 2022).

3.3.3 *Visual Studio*

O Microsoft *Visual Studio* é um ambiente de desenvolvimento integrado (IDE) da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e F# (F Sharp). Também é um produto de desenvolvimento na área web, usando a plataforma do ASP.NET, como websites, aplicativos web, serviços web e aplicativos móveis. Atualmente, o *Visual Studio* é uma das IDEs mais relevantes no mercado. Isso se deve aos vários recursos que essa ferramenta disponibiliza (MICROSOFT. 2022).

Uma das principais vantagens da utilização do *Visual Studio* é que o mesmo é um assistente para desenvolvimento, ou seja, independentemente da linguagem utilizada, ele mostra os atalhos de APIs disponíveis, além de fazer preenchimento automático dos comandos para agilizar a construção do código. Através dele também é possível depurar o código com intuito de verificar erros de sintaxe e da lógica de comandos do programa (BLOG IMPACTA, 2017).

4 DESENVOLVIMENTO

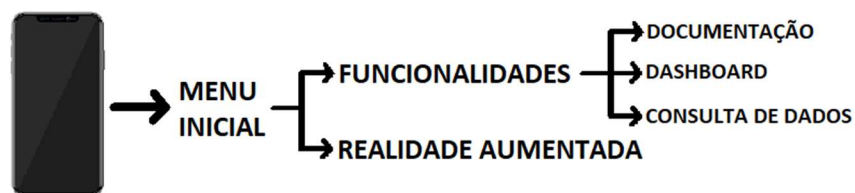
Para o desenvolvimento do presente trabalho, buscando atingir os objetivos propostos, elencou-se as prioridades a serem desenvolvidas e definiu-se um escopo a ser seguido baseando-se na metodologia proposta.

4.1 Metodologia

Uma vez conhecidas as ferramentas para a elaboração deste trabalho e os objetivos a serem atingidos foi possível determinar a metodologia utilizada no desenvolvimento da solução, para isso estipulou-se uma estrutura de processos de metodologia ágil, seguindo um desenvolvimento iterativo e uma arquitetura baseada em componentes. Optou-se por este modelo devido a diminuição dos riscos que a estrutura propõe visto que acompanha o progresso de perto, realizando uma avaliação do que foi feito a cada etapa e possui foco em subsistemas que apresentem funções muito bem definidas.

O planejamento inicial estabeleceu o comportamento e os dispositivos compatíveis com a aplicação, além de quais seriam os requisitos principais para o sistema e a definição do escopo do projeto para um direcionamento às etapas posteriores. Desta forma optou-se por uma divisão por iterações, ou seja, por loops de desenvolvimento que resultam em uma entrega e chegou-se no primeiro esboço da solução como mostra a Figura 7.

Figura 7 - Esquemático da solução.



Fonte: Autoria própria.

Assim, a fim de atingir os objetivos estabelecidos, projetou-se um aplicativo contendo cinco telas, sendo uma delas o menu inicial, responsável pela centralização das funcionalidades, e as outras quatro responsáveis por:

- Documentação - Centralizar toda a documentação referente à bancada, de modo a facilitar a procura pelos mesmos, além de permitir a abertura dos documentos por dentro do aplicativo sem a necessidade da abertura de um outro aplicativo;
- Dashboard - Integração com o dashboard de dados presente na IHM da planta, desenvolvido em Domingues (2021), possibilitando assim a visualização de todos os dados referentes ao processo em tempo real e sem a necessidade de estar por perto da planta;
- Assistente Virtual – Integração com assistente virtual para automação da consulta na base de dados através de PLN (Processamento de Língua Natural), desenvolvido em Fernandez (2021), permitindo o acesso as informações importantes de forma simples e eficaz;
- Realidade Aumentada - Monitoramento do processo através de AR com exibição dos dados em tempo real de forma conjunta (entradas e saídas) além da possibilidade da visualização por cada sensor e a abertura dos respectivos datasheets

Todo o desenvolvimento foi feito através do *Unity* em conjunto com o IDE (*Integrated Development Environment*) *Visual Studio* para a codificação. Para o desenvolvimento das funcionalidades utilizou-se do plug-in do *UniWebView*, pois o mesmo permite a abertura de páginas web por dentro do aplicativo, enquanto para o desenvolvimento da AR utilizou-se do *Vuforia* e de suas funcionalidades devido a facilidade de integração com o *Unity* e a simplicidade na utilização.

A partir deste ponto foi possível preparar o *Unity* para a criação do aplicativo de AR, para isso houve a importação e instalação dos SDKs *Android* e *Vuforia*, assim como também o JDK (*Java Development Kit*).

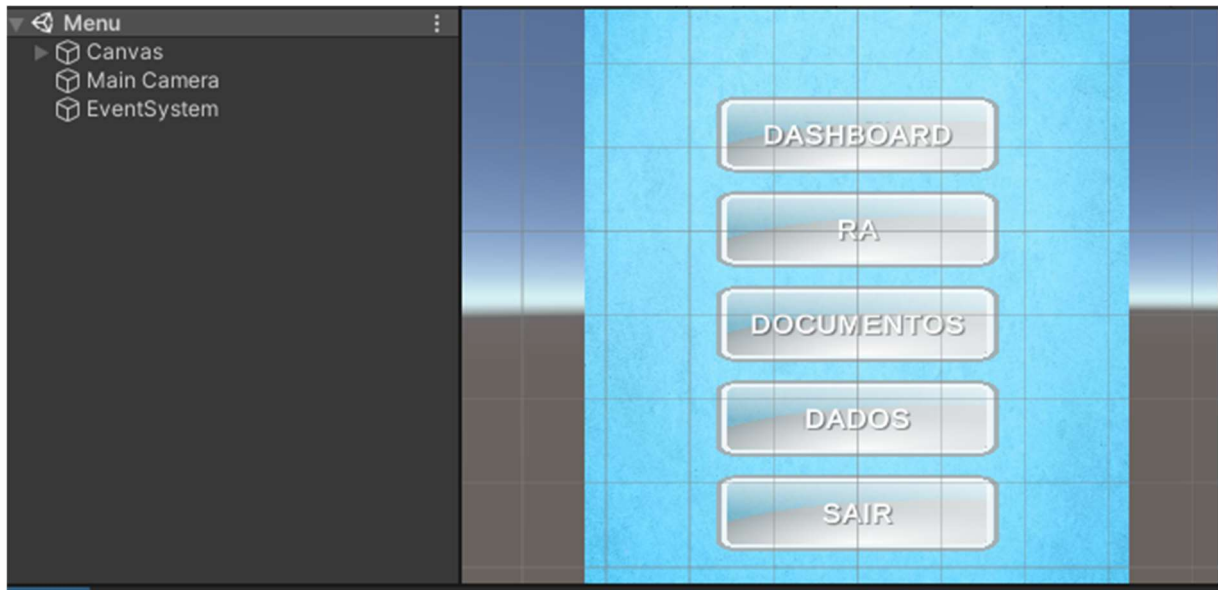
Finalizando-se a preparação, todas as funcionalidades e componentes foram desenvolvidos e integrados ao aplicativo e por fim realizou-se os testes para identificação de falhas na aplicação e gerou-se o produto final disponibilizando o mesmo para o usuário final.

4.2 Menu Inicial do Aplicativo

Para o menu inicial definiu-se uma tela simples e funcional, onde o usuário possa ter acesso a todas as funcionalidades do aplicativo de forma clara e intuitiva visto que o aplicativo é voltado para aplicações industriais e, desta forma, a operabilidade do mesmo deve ser garantida. Deste modo, projetou-se a tela inicial do aplicativo com quatro botões em que cada um representa e abre uma funcionalidade diferente.

A opção por iniciar pela tela inicial se deu de modo a garantir uma prioridade de cenas na *Unity*, onde a mesma torna-se a cena 0 do projeto e assim garantindo a ser a primeira na abertura do aplicativo. Após definido o modelo e o projeto, iniciou-se o desenvolvimento da tela no *Unity*, para isso criou-se um novo Canvas e adicionou-se os quatros botões das funcionalidades e um para sair do aplicativo, como mostra a Figura 8.

Figura 8 - Tela do menu inicial.



Fonte: Autoria própria.

Finalizando o visual, iniciou-se o desenvolvimento do script em C# responsável pela abertura das telas. A Figura 9 exemplifica o script desenvolvido, através dela é possível notar a utilização da função *LoadScene* juntamente com o nome da cena. Por se tratar de uma engine de jogos as telas do aplicativo são divididas como se fossem diferentes cenas de um jogo.

Figura 9 - Script do menu inicial.

```

public class MainMenu : MonoBehaviour
{
    public void PlayRA()
    {
        SceneManager.LoadScene("Example");
    }

    public void PlayDocs()
    {
        SceneManager.LoadScene("Docs");
    }

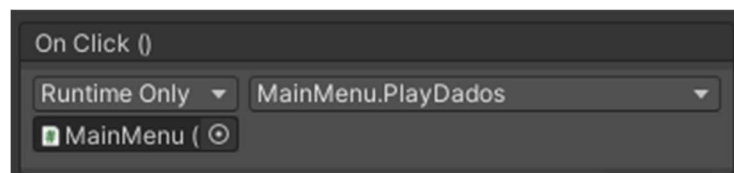
    public void PlayDados()
    {
        SceneManager.LoadScene("Dados");
    }

    public void Sair()
    {
        Debug.Log("QUIT!");
        Application.Quit();
    }
}

```

Fonte: Autoria própria.

Por fim, realizou-se a configuração dos botões, onde foi importado o script para dentro da função do clique e selecionou-se qual função do script é executada ao clicar no botão, como exemplifica a Figura 10.

Figura 10 - Função do botão no menu inicial.

Fonte: Autoria própria.

4.3 Realidade Aumentada

Por meio da utilização da realidade aumentada definiu-se que seja possível realizar o monitoramento do processo através da visualização em tempo real dos valores medidos pelos sensores, para isso dividiu-se o desenvolvimento em partes como mostra a Figura 11:

Figura 11 - Fluxo do desenvolvimento.

Fonte: Autoria própria.

Conforme exemplificado na Figura 11, primeiramente estipulou o tipo de *target* que seria utilizado e desenvolveu-se o mesmo através do auxílio do *Vuforia*. Finalizando o *target* iniciou-se o desenvolvimento da parte visual da realidade aumentada mediante a utilização da plataforma *Unity*, com integração a plataforma *Vuforia*, e por fim foi desenvolvida a comunicação através de requisições REST, via uma API padronizada disponibilizada pelo framework *Molecular*, em que busca-se apresentar os valores das variáveis, bem como a situação em tempo real das malhas de processo da bancada usando AR.

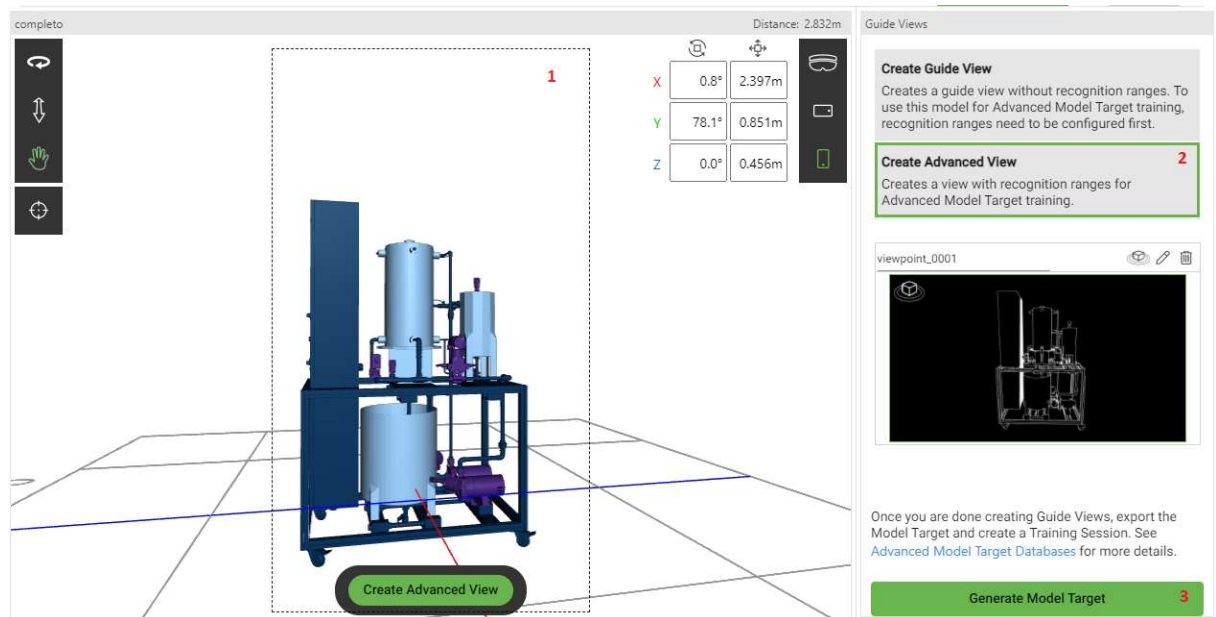
4.3.1 Target

O desenvolvimento do *target* iniciou-se pela definição do mesmo, optou-se pela utilização na forma de *Model Target Based* (Alvo baseado em um modelo), pois este modelo permite que o aplicativo reconheça e rastreie objetos específicos no mundo real com base na forma do mesmo. Sendo assim, ao identificar-se a bancada de processos, que foi definida como alvo, inicia-se o processo de realidade aumentada. Este processo é permitido graças ao *Model Target Generator* disponibilizado pela *Vuforia*. Através do *Model Target Generator* é possível gerar

de forma simples o *target*, a Figura 12 exemplifica o processo de criação do alvo através do *Model Target Generator*.

Primeiramente, importou-se o desenho 3D da bancada para dentro do *software* e juntamente foi informando as proporções do mesmo (1) e, em seguida, para que fosse possível a utilização e de modo a informar qual seria a delimitação para utilização do AR, estipulou-se o *viewgrid* (2) e gerou-se o banco de dados (3) que posteriormente é importado na *Unity*.

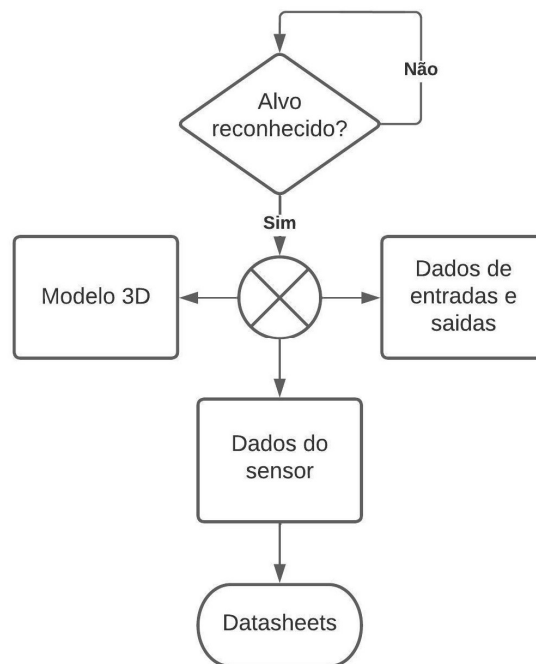
Figura 12 - Geração do *Model Target*.



Fonte: Autoria própria.

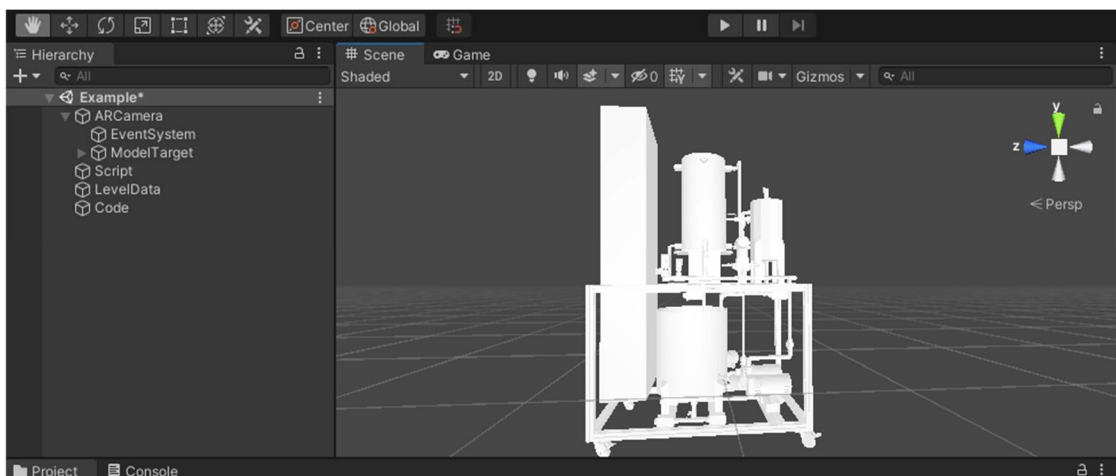
4.3.2 Desenvolvimento Visual

Após o desenvolvimento do *target* se deu início a criação da parte visual da realidade aumentada, nesta etapa desenvolveu-se todo o visual que será exibido durante a ativação da AR. Para isso estipulou-se que ao reconhecer-se o alvo seria exibido um modelo 3D sob a planta e juntamente com o modelo seria possível observar os valores de entrada e saída em tempo real, bem como o valor medido por cada sensor de forma independente junto com a possibilidade de abertura do respectivo datasheet. A Figura 13 exemplifica o processo visual do aplicativo.

Figura 13 - Fluxo visual do aplicativo.

Fonte: Autoria própria.

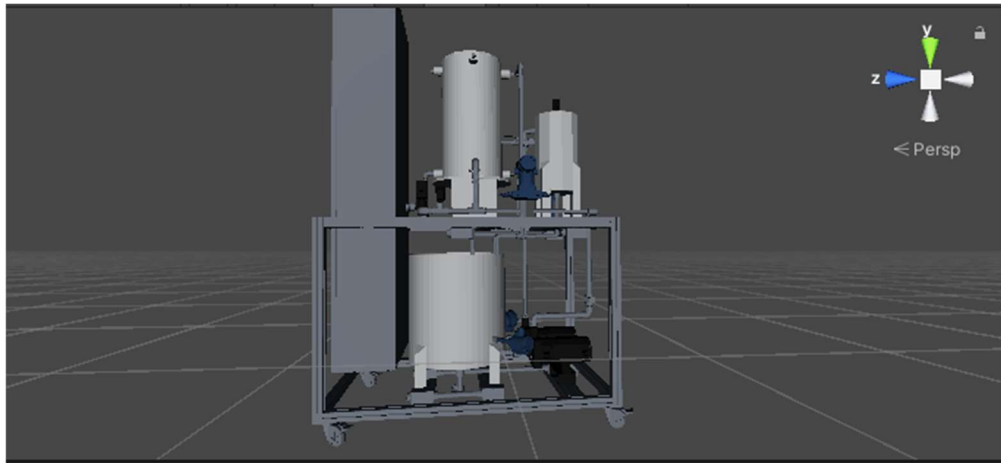
Inicialmente configurou-se o *Unity* para que fosse possível o desenvolvimento e utilização da realidade aumentada, desta forma foi configurado o desenvolvimento compatível com plataforma *Android* e foi habilitado o *Vuforia* Realidade Aumentada. Em seguida importou-se os pacotes de realidade aumentada do *Vuforia*, o banco de dados gerado pelo MTG e definiu-se o mesmo como referência para a câmera. A Figura 14 exemplifica o resultado obtido.

Figura 14 - Importação do MTG no *Unity*.

Fonte: Autoria própria.

Finalizado as configurações iniciais foi possível dar início ao desenvolvimento visual do aplicativo. Primeiramente, importou-se para dentro do *Unity* o modelo 3D renderizado que é exibido ao identificar-se o alvo e adicionou-se o mesmo a cena, alinhando-o com o *viewgrid* gerado pelo MTG de modo a atingir a exibição correta. A Figura 15 exemplifica o resultado obtido.

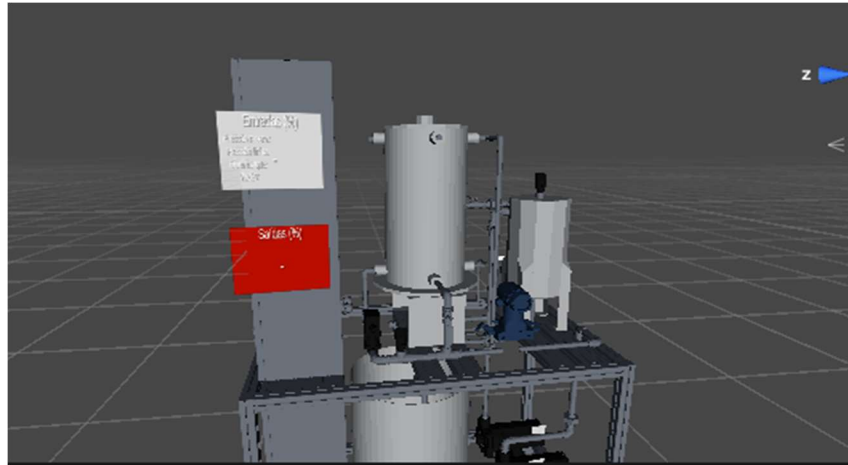
Figura 15 - Início do desenvolvimento visual no *Unity*.



Fonte: Autoria própria.

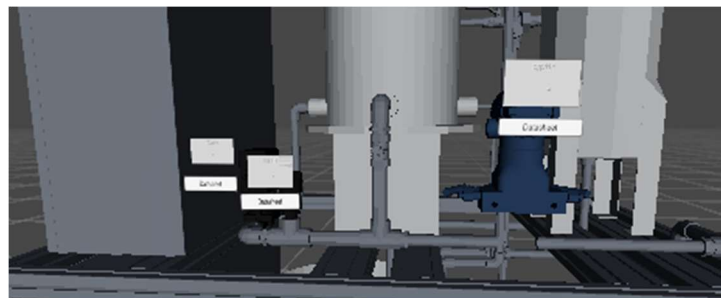
Após finalizada a referência, iniciou-se o desenvolvimento dos objetos responsáveis pela visualização dos dados durante a exibição da realidade aumentada. Para isto criou-se um canva e, primeiramente, desenvolveu-se os objetos responsáveis pela visualização das entradas e saídas no geral como mostra a Figura 16. Posteriormente, de semelhante modo, desenvolveu-se os objetos para a visualização e os botões de datasheets de cada sensor respectivamente como mostra a Figura 17.

Figura 16 - Desenvolvimento visual no *Unity*.



Fonte: Autoria própria.

Figura 17 - Desenvolvimento visual por sensor no *Unity*.



Fonte: Autoria própria.

4.3.3 Comunicação com a API

Os dados da planta são disponibilizados através de uma API, o processo para obtenção dos dados inicia-se com o microsserviço de aquisição de dados, chamado de DAQ, que faz a leitura dos sensores, controladores e atuadores da planta piloto, esses dados são disponibilizados para o Serviço denominado Influx através de uma API via protocolo HTTP. Entre o DAQ e a API é utilizado o *transporter* para trocar informações (PONTAROLLI, 2020).

Para o protocolo HTTP, tecnologia utilizada para fazer as requisições dos dados, são necessários conhecer as rotas, que são endereços de rede chamadas de URL (sigla para o termo em inglês *Uniform Resource Locator*) disponibilizadas pelos microsserviços para consulta dos dados (PONTAROLLI, 2020). Fazendo-se uma consulta, cada uma destas rotas retornam um conjunto de valores no formato JSON. As rotas utilizadas no presente trabalho possuem três ações: as duas primeiras são referentes aos sensores e a terceira à atuadores. Na Tabela 1 estão relacionadas as URLs completas e as ações além dos componentes a que se referem.

Tabela 1- Endereços requisitados pelo serviço *Influx*.

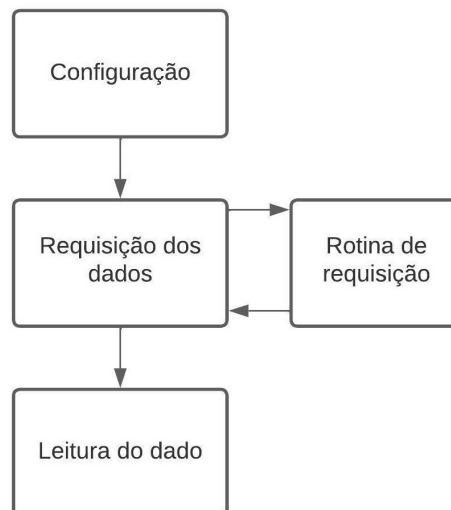
	Rota	Ações	Componentes
1	http://192.168.1.201:3005/api/daq/riin	riin	Sensores
2	http://192.168.1.201:3005/api/daq/ruin	ruin	Sensores
3	http://192.168.1.201:3005/api/daq/ruout	ruout	Atuadores

Fonte: Autoria própria.

4.3.3.1 Requisição dos Dados de Variáveis do Processo

Conhecendo-se o funcionamento da comunicação e requisição de dados foi possível iniciar-se o desenvolvimento da requisição de dados. O desenvolvimento é exemplificado na Figura 18, onde é possível notar que optou-se pela criação em métodos separados para facilitar em possíveis mudanças que surgissem ao longo do desenvolvimento e também para se ter uma maior facilidade ao indicar-se erros ao longo do processo de criação.

O desenvolvimento pode ser dividido, de forma primária, em três partes: inicialmente realiza-se a configuração da comunicação, sendo esta responsável por conter as URL's que serão requisitadas, após a configuração realiza-se a requisição dos dados e por fim a leitura e interpretação dos mesmos.

Figura 18 - Primeira parte da comunicação.

Fonte: Autoria própria.

Primeiramente criou-se uma classe chamada *ServerConfig* para organizar as chaves e strings utilizadas nas chamadas da API. Optou-se por esta estratégia para que caso fosse necessário alterar caminhos, ou URLs, para API não seja necessária alteração no código da chamada e sim nesta classe à parte. A Figura 19 exemplifica a classe criada.

Figura 19 - Classe de configuração da comunicação.

```
public class ServerConfig
{
    public const string SERVER_API_URL_FORMAT = "http://192.168.1.201:3005/api/da
    q/{0}";

    public const string API_GET_QUOTE = "riin";
    public const string API_GET_OUTPUT = "ruout";
}
```

Fonte: Autoria própria.

Após o desenvolvimento da classe iniciou-se o desenvolvimento da requisição dos dados, para isto, utilizou-se da função *SendRequest* pois a mesma permite fazer requisições GET de forma flexível e passar a resposta do servidor para um *callback* que, posteriormente, foi possível processar em outras funções (SLIDE FACTORY, 2020). Juntamente com a função utilizou-se de uma co-rotina como mostra a Figura 20.

Figura 20 - Função SendRequest.

```
/// <summary>
/// This method is used to begin sending request process.
/// </summary>
/// <param name="url">API url.</param>
/// <param name="callbackOnSuccess">Callback on success.</param>
/// <param name="callbackOnFail">Callback on fail.</param>
/// <typeparam name="T">Data Model Type.</typeparam>
private void SendRequest<T>(string url, UnityAction<T> callbackOnSuccess,
    UnityAction<string> callbackOnFail)
{
    StartCoroutine(RequestCoroutine(url, callbackOnSuccess,
        callbackOnFail));
}
```

Fonte: Autoria própria.

No *Unity*, uma co-rotina é um método que pode pausar a execução e retornar o controle ao *Unity*, mas continuar de onde parou no quadro seguinte. Na maioria das situações, ao chamar-se um método, o mesmo é executado até a conclusão e, em seguida, retorna o controle ao método chamador, além de quaisquer valores de retorno opcionais. Isso significa que qualquer ação que ocorra em um método deve ocorrer em uma única atualização de quadro (UNITY TECHNOLOGIES, 2022).

Para um maior controle do método é utilizado as co-rotinas, pois elas são executadas de maneira diferente no *Unity*. O compilador C# automaticamente gera uma instância de uma classe que suporta co-rotinas e o *Unity* usa esse objeto para rastrear o estado da co-rotina em várias invocações de um único método. Desta forma, o *Unity* eleva as variáveis de escopo local na classe gerada, que permanece alocada durante a co-rotina e permite rastrear o estado interno da co-rotina, lembrando assim em qual ponto do código a co-rotina deve ser retomada após a entrega (UNITY TECHNOLOGIES, 2022).

Desta forma, a requisição de dados é feita dentro da co-rotina juntamente com a utilização do método *UnityWebRequest*. O método *UnityWebRequest* fornece um sistema modular para compor solicitações HTTP e manipular suas respostas. O objetivo principal do sistema é permitir que aplicativos desenvolvidos pela *Unity* interajam com back-ends da web modernos (UNITY TECHNOLOGIES, 2022).

De forma resumida a co-rotina envia a requisição com a URL enviada e devolve ao programa principal o resultado sendo ele um erro ou não, como exemplifica a Figura 21.

Figura 21 - Co-rotina criada.

```
private IEnumerator RequestCoroutine<T>(string url, UnityAction<T>
    callbackOnSuccess, UnityAction<string> callbackOnFail)
{
    var www = UnityWebRequest.Get(url);
    www.certificateHandler = new BypassCertificate();
    yield return www.SendWebRequest();

    if (www.isNetworkError || www.isHttpError)
    {
        Debug.LogError(www.error);
        callbackOnFail?.Invoke(www.error);
    }
    else
    {
        Debug.Log(www.downloadHandler.text);
        ParseResponse(www.downloadHandler.text, callbackOnSuccess,
            callbackOnFail);
    }
}
```

Fonte: Autoria própria.

Finalizada a co-rotina de requisição, criou-se um método para a finalização do processo e a leitura dos dados. O dado lido através da requisição é recebido em formato JSON, desta forma para que o mesmo possa ser interpretado mais a frente e não fique preso na co-rotina de requisição criou-se a função *ParseResponse*, como mostra a Figura 22.

Figura 22 - Finalização da comunicação.

```

/// </summary>
/// <param name="data">Data received from server in JSON format.</param>
/// <param name="callbackOnSuccess">Callback on success.</param>
/// <param name="callbackOnFail">Callback on fail.</param>
/// <typeparam name="T">Data Model Type.</typeparam>
private void ParseResponse<T>(string data, UnityAction<T>
    callbackOnSuccess, UnityAction<string> callbackOnFail)
{
    var parsedData = JsonUtility.FromJson<T>(data);
    callbackOnSuccess?.Invoke(parsedData);
}

```

Fonte: Autoria própria.

Finalizado o desenvolvimento da comunicação, optou-se por criar funções para requisição das entradas (sensores) e saídas (atuadores) de modo a facilitar o processo mais a frente. Desta forma utilizou-se dos métodos criados ao longo do desenvolvimento, contudo já passando os parâmetros definidos sem que seja necessário passar novamente para a requisição em tempo real. A Figura 23 exemplifica as funções criadas.

Figura 23 - Métodos para requisição de entradas e saídas.

```

/// <summary>
/// This method call server API to get a quote.
/// </summary>
/// <param name="callbackOnSuccess">Callback on success.</param>
/// <param name="callbackOnFail">Callback on fail.</param>
public void GetQuote(UnityAction<Root> callbackOnSuccess,
    UnityAction<string> callbackOnFail)
{
    SendRequest(string.Format(ServerConfig.SERVER_API_URL_FORMAT,
        ServerConfig.API_GET_QUOTE), callbackOnSuccess, callbackOnFail);
}

/// <summary>
/// This method call server API to get a quote.
/// </summary>
/// <param name="callbackOnSuccess">Callback on success.</param>
/// <param name="callbackOnFail">Callback on fail.</param>
public void GetOutput(UnityAction<Roup> callbackOnSuccess,
    UnityAction<string> callbackOnFail)
{
    SendRequest(string.Format(ServerConfig.SERVER_API_URL_FORMAT,
        ServerConfig.API_GET_OUTPUT), callbackOnSuccess, callbackOnFail);
}

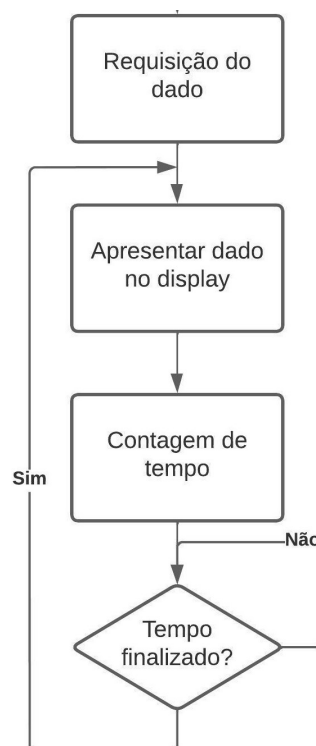
```

Fonte: Autoria própria.

4.3.3.2 Comunicação em Tempo Real

Finalizando-se a comunicação focou-se na visualização dos dados em tempo real, primeiramente definiu-se qual seria o método utilizado para que a requisição acontecesse em tempo real, a partir de estudos optou-se pela utilização de um *countdown* de tempo em que a cada *loop* o tempo era decrescido até que ao chegar a zero o processo se repete. A Figura 24 exemplifica o funcionamento da comunicação em tempo real.

Figura 24 - Fluxo da segunda parte da comunicação.



Fonte: Autoria própria.

Após ter definido o método desenvolveu-se o script para a requisição em tempo real, desta forma, criou-se inicialmente um *start* que chama a função *CallAPI*, desenvolvida posteriormente, e em seguida gerou-se uma função *update* para a atualização. O *update* utiliza-se da função *Time.deltaTime*, onde o *deltaTime* representa o intervalo de tempo entre uma atualização e outra. Desta forma a função *CallAPI* é chamada a cada 1,5 segundos já que o *API_TIME* foi definido em 1.5f. A Figura 25 exemplifica a requisição.

Figura 25 - Script para requisição em tempo real

```

public class UIServerFIT126Button : MonoBehaviour
{
    // Reference to label in scene.
    [SerializeField]
    private UIServerResponseViewer4 display4;
    private const float API_TIME3 = 1.5f;
    private float apiCheckCountdown3 = API_TIME3;
    // private string dado4;

    private void Start()
    {
        CallAPI3();
    }
    private void Update()
    {
        apiCheckCountdown3 -= Time.deltaTime;
        if (apiCheckCountdown3 <= 0)
        {
            CallAPI3();
            apiCheckCountdown3 = API_TIME3;
        }
    }
}

```

Fonte: Autoria própria.

Para a função *CallAPI* utilizou-se do método de chamada de eventos onde realiza-se a comunicação com o servidor, já explicada anteriormente, e de acordo com o resultado chama-se uma função diferente. Em caso de sucesso na comunicação é mostrado no display, através da função *ShowMessage* explicada mais à frente, contudo, em caso de não haver comunicação é exibido no display o erro resultante de modo que desta forma gera-se uma maior facilidade na busca da solução do mesmo. A Figura 26 exemplifica a função criada.

Figura 26 - Função *CallAPI*.

```

/// <summary>
/// Method called on event.
/// </summary>
public void CallAPI3()
{
    ServerCommunication.Instance.GetQuote(APICallSucceed3, APICallFailed3);
}

/// <summary>
/// Request was successful
/// </summary>
/// <param name="quoteData">Quote data.</param>
private void APICallSucceed3(Root quoteData)
{
    display4.ShowMessage4(string.Format("{3:N3}%", quoteData.riin));
}

/// <summary>
/// There were some problems with request.
/// </summary>
/// <param name="errorMessage">Error message.</param>
private void APICallFailed3(string errorMessage)
{
    display4.ShowMessage4(string.Format("Error\n{0}", errorMessage));
}
}

```

Fonte: Autoria própria.

A função *ShowMessage* é utilizada para o controle do *Label*, em que cada componente possui um *Label* diferente. Optou-se pela utilização desta função para que fosse possível realizar os controles dos displays de uma forma mais simples e eficiente, a função indica para o *Unity* onde deve ser exibido o resultado e desta forma garante que não haverá resultados em displays errados. A Figura 27 exemplifica a função criada.

Figura 27 - Função ShowMessage.

```

/// <summary>
/// This class is responsible for label control.
/// </summary>
public class UIServerResponseViewer3 : MonoBehaviour
{
    // Reference to label.
    [SerializeField]
    private TextMeshProUGUI label3;

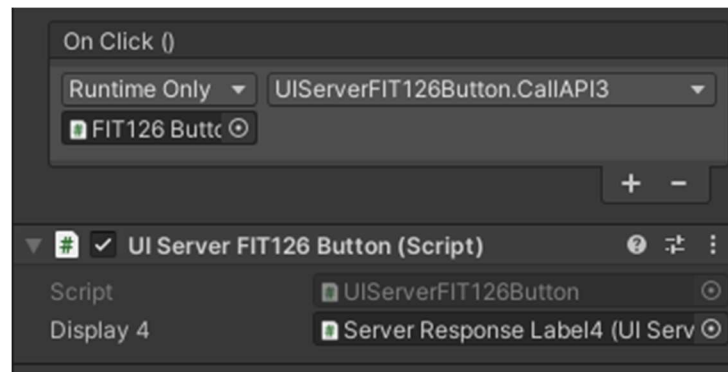
    /// <summary>
    /// Method used to display message on screen.
    /// </summary>
    /// <param name="message">Message.</param>
    public void ShowMessage3(string message)
    {
        label3.text = message;
    }

    /// <summary>
    /// Method used to clear label content.
    /// </summary>
    public void ClearMessage3()
    {
        label3.text = string.Empty;
    }
}

```

Fonte: Autoria própria

Concluída a segunda parte da comunicação iniciou-se a configuração da mesma dentro da plataforma do *Unity*, nesta etapa cada display, que foi previamente alocado na frente do respectivo sensor, recebeu o script correspondente ao sensor representado e foi selecionado o label de exibição, como mostra a Figura 28.

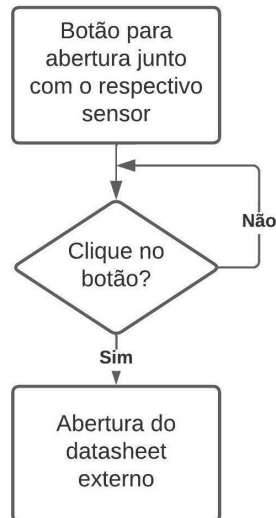
Figura 28 - Configuração do scrip e label.

Fonte: Autoria própria.

Como é possível notar na Figura 26, para cada botão foi importado o respectivo script e designado um display para ser responsável pelo label de exibição dos dados, onde utilizou-se do pacote *TextMeshPro*, pois o mesmo permite um controle maior sobre a fonte e forma de exibição. Juntamente, também definiu-se o chamado da função *CallAPI* e a mesma é atualizada em tempo real conforme explicado anteriormente.

4.3.4 Datasheets

Por fim, iniciou-se o desenvolvimento do Script responsável pela visualização do datasheet de cada sensor. O objetivo dessa funcionalidade é permitir que o usuário abra o documento diretamente da tela da realidade aumentada e assim facilitar no processo de tomada de decisão no caso de notar-se valores medidos fora do esperado. Para isso, a visualização é feita através do navegador web e de forma independente do aplicativo. Optou-se por esta forma para que o aplicativo não ficasse com tamanho excessivo. A Figura 29 exemplifica o processo de abertura dos datasheets.

Figura 29 - Fluxo de funcionamento da abertura dos datasheets.

Fonte: Autoria própria.

O Script é baseado na URL de cada datasheet, que foi retirada do site do fornecedor, onde ao ser clicado no botão do datasheet ele envia uma solicitação para abrir a URL no navegador web. A solicitação é feita através da função padrão *Application.OpenURL*, esta função abre a URL especificada no navegador padrão e trás o aplicativo do mesmo para a frente, a Figura 30 exemplifica o script criado.

Figura 30 - Scrip datasheets

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class WebURLScript : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    public void btnOne()
    {
        Application.OpenURL("https://www.emerson.com/documents/automation/
        ficha-de-dados-do-produto-rosemount-2051-transmissor-de-press%E3o-
        pt-89044.pdf");
    }

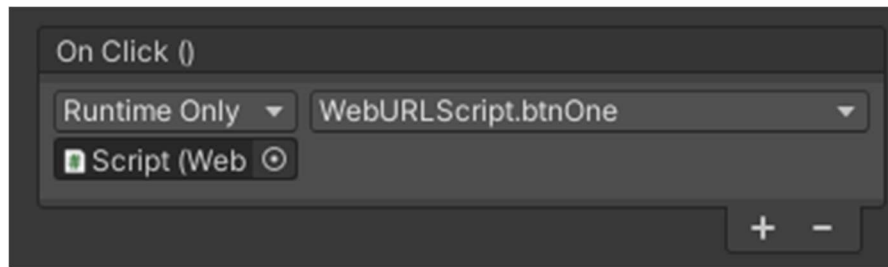
    public void btnThree()
    {
        Application.OpenURL("https://www.emerson.com/documents/automation/
        ficha-de-dados-do-produto-rosemount-2051-transmissor-de-press%E3o-
        pt-89044.pdf");
    }

    public void btnFour()
    {
        Application.OpenURL("http://www.dancor.com.br/dancor-site-novo/public/
        uploads/produtos/dancor%20pratika/cat%C3%A1logos/cp-4r-pbe_cat.pdf");
    }
}
  
```

Fonte: Autoria própria.

Após o desenvolvimento do script foi feita a configuração dos botões na *Unity* para que cada botão abrisse o datasheet do sensor que o mesmo representa. Primeiramente importou-se o script para o botão e posteriormente, como mostra a Figura 31 abaixo, selecionou-se qual ação deveria ser feita no clique do botão.

Figura 31 - Configuração botão do datasheets.



Fonte: Autoria própria.

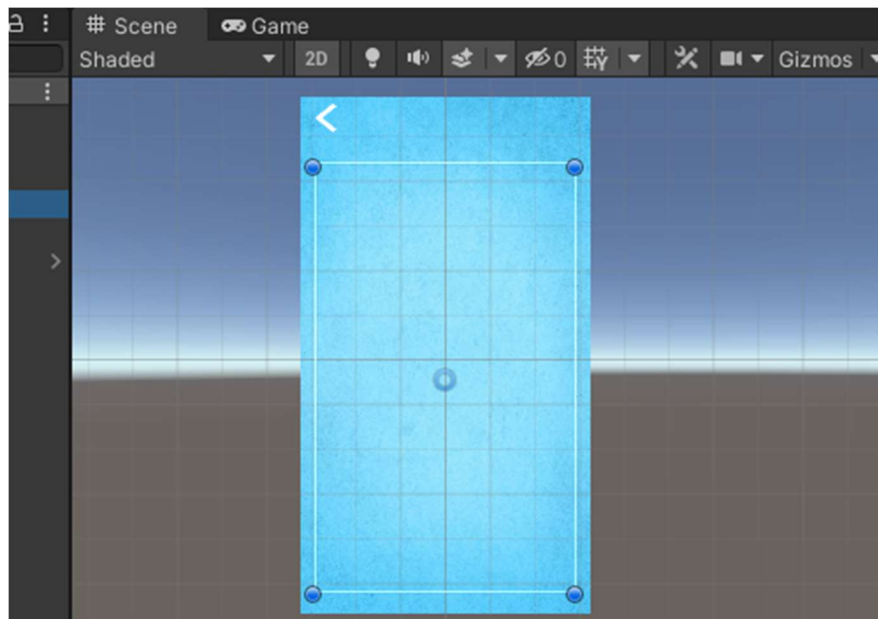
4.4 Funcionalidades

Para as funcionalidades que utilizam páginas na web, como o dashboard, o conjunto da documentação e a consulta de dados, optou-se pela utilização do plug-in UniWebView com o objetivo de atingir uma maior fluidez e ao mesmo tempo garantindo a operabilidade do aplicativo.

Através do UniWebView é possível abrir páginas web por dentro do aplicativo sem que seja necessário a abertura do *browser* além da possibilidade da utilização de um Prefab que facilita o desenvolvimento. O Prefab é um tipo especial de componente que permite que *GameObjects* totalmente configurados sejam salvos no projeto para reutilização e desta forma não é necessário realizar a configuração novamente.

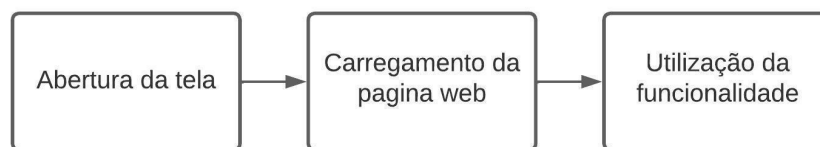
Desta forma, conforme exemplificado na seção 4.1, elaborou-se as três telas de funcionalidades seguindo o mesmo padrão de desenvolvimento, como mostra a Figura 32, onde criou-se um novo canva e importou-se o Prefab do UniWebView alterando somente a configuração de abertura de tela. A Figura 33 exemplifica o fluxo de funcionamento.

Figura 32 - Tela padrão das funcionalidades.



Fonte: Autoria própria.

Figura 33 - Fluxo de funcionamento das funcionalidades.

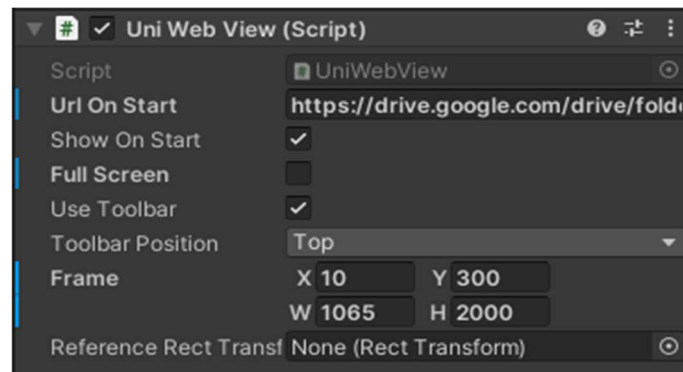


Fonte: Autoria própria.

4.4.1 Datasheets e Documentação

A primeira funcionalidade desenvolvida foi a de abertura dos datasheets e documentação. A configuração feita no prefab do UniWebView é mostrada na Figura 34.

Figura 34 - UniWebView para abertura dos documentos.



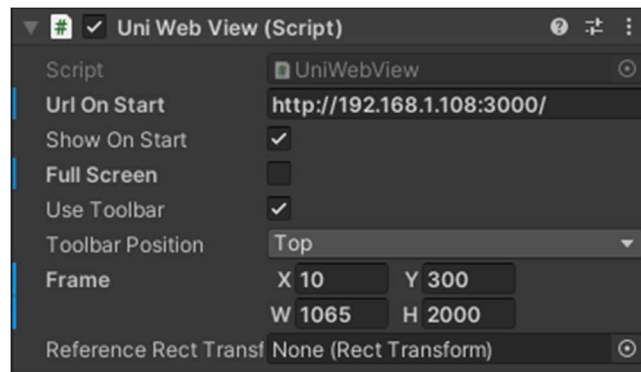
Fonte: Autoria própria.

Como é possível notar, foi realizada a configuração para a abertura de uma pasta do Google Drive. A opção pela utilização de uma pasta do Google Drive se deu pelo fato de que desta forma a inclusão de novos documentos no aplicativo se dá pelo mesmo, sem a necessidade de alteração no código fonte.

4.4.2 Dashboard

A segunda funcionalidade desenvolvida foi a integração com o dashboard de dados presente na IHM da planta, desenvolvido em Domingues (2021), possibilitando assim a visualização de todos os dados referentes ao processo em tempo real e sem a necessidade de estar por perto da planta. O dashboard contém as informações dos sensores, atuadores e malhas de processo da planta piloto, possibilitando assim a visualização dos dados sem a necessidade do reconhecimento da Realidade Aumentada. A Figura 35 exemplifica a configuração do prefab do UniWebView realizada.

Figura 35 - UniWebView para abertura do dashboard.



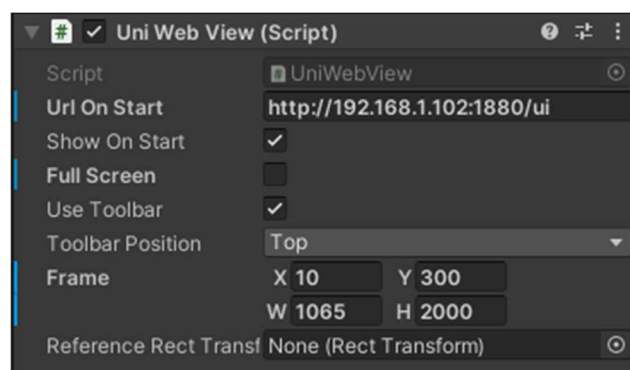
Fonte: Autoria própria.

4.4.3 Consulta de dados

Por fim, implementou-se integração com o assistente virtual para automação de consulta na base de dados da planta piloto através de PLN (Processamento de Linguagem Natural), desenvolvido em Fernandez (2021), permitindo assim acesso a informações importantes de forma simplificada, onde a consulta se dá através da digitação da pergunta e após interpretação da mesma é apresentada a resposta buscada no banco de dados.

O presente trabalho não trata o desenvolvimento da PLN e sim da integração do mesmo dentro do aplicativo desenvolvido para que fosse possível centralizar todas as funcionalidades referentes a bancada de processos em um só lugar. A Figura 36 exemplifica a configuração para a abertura da rota correspondente ao PLN.

Figura 36 - UniWebView para abertura do PLN.



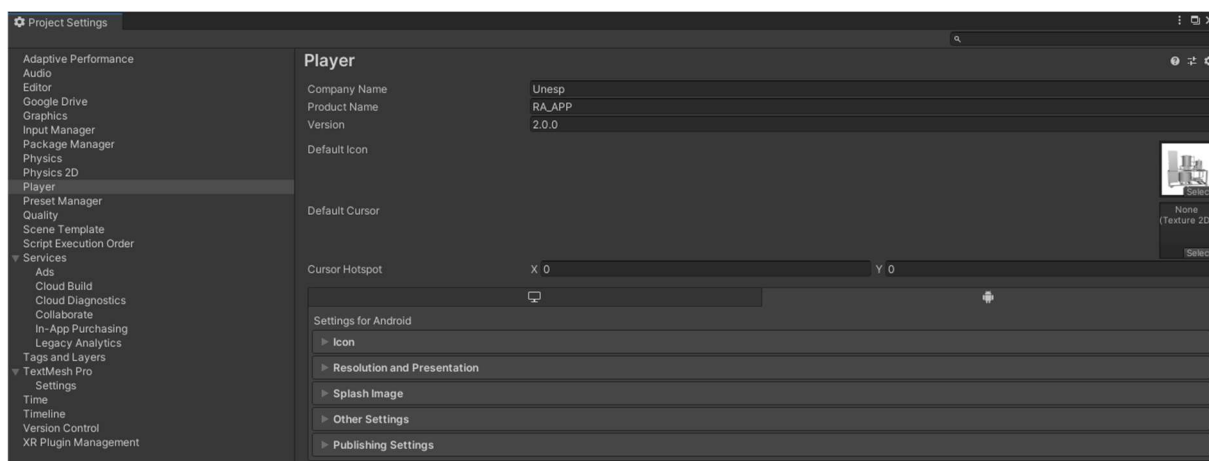
Fonte: Autoria própria.

4.5 Geração do Aplicativo *Android* (APK)

O *Unity* permite realizar diversas configurações que se referem ao projeto dando assim uma maior flexibilidade ao desenvolvimento e maior controle sobre o produto final. Diante destas possibilidades, após finalizado a comunicação e o visual, iniciou-se as configurações do projeto, inicialmente habilitou-se a utilização do protocolo HTTP (*Hypertext Transfer Protocol*) pelo UniWebView devido ao *Android* não permitir mais a utilização do mesmo, apenas HTTPS, contudo as rotas para o acesso das funcionalidades utilizam deste protocolo e sem habilitar a função não seria possível realizar a comunicação.

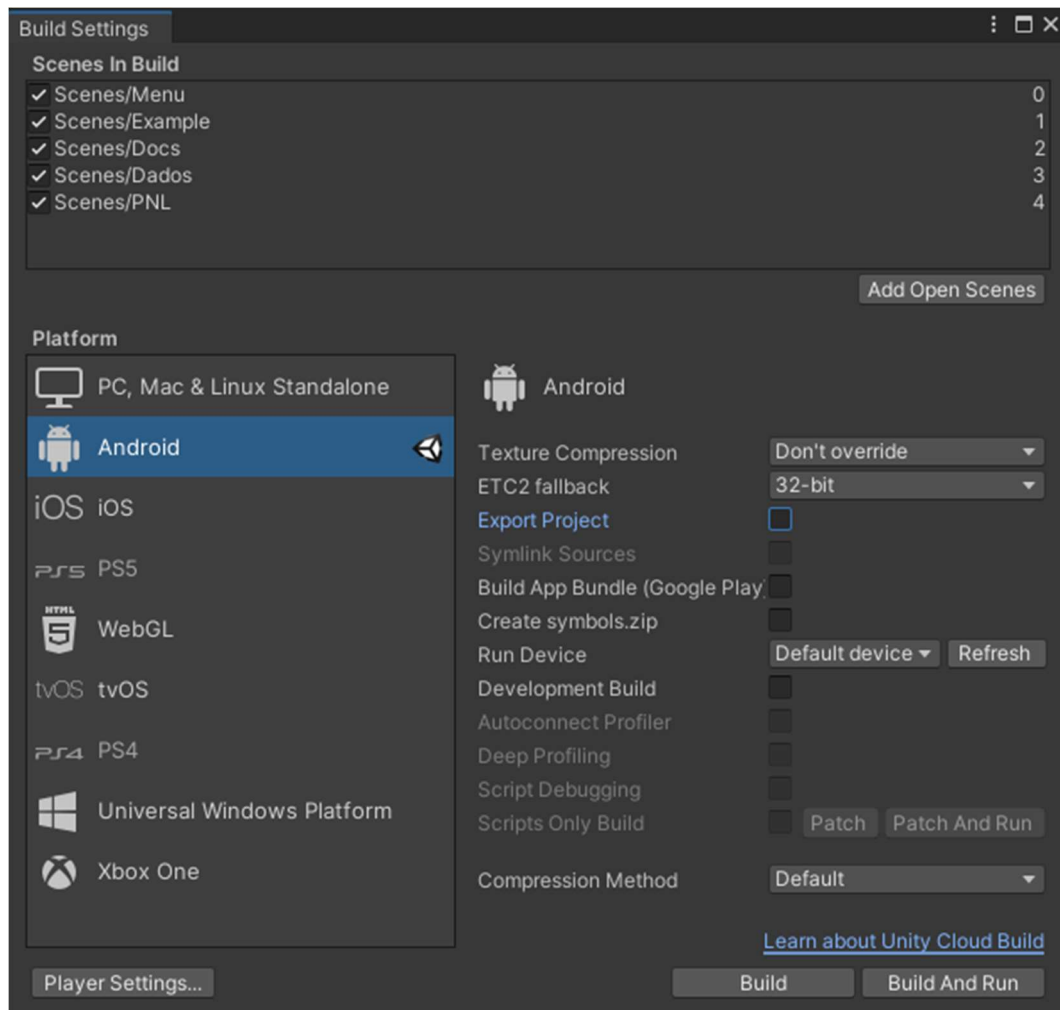
Posteriormente, através da aba *Player Settings*, o *Unity* possibilita configurações referentes a utilização no *Android* como por exemplo definição da versão mínima, resolução, entre outros. Para o presente trabalho não entrou-se em quesitos mais avançados e configurou-se apenas definições como versão, nome, ícone, entre outras, conforme mostrado na Figura 37.

Figura 37 - Configuração do APK.



Fonte: Autoria própria.

O *Unity* fornece uma ferramenta que permite gerar o aplicativo final diretamente por dentro do *software* sem que seja necessário a instalação de ferramentas adicionais. Desta forma, selecionou-se todas as cenas geradas ao longo do desenvolvimento e gerou-se o apk compatível com *Android* como mostra a Figura 38. Nota-se também que o *Unity* oferece diversas outras possibilidades de sistemas operacionais, permitindo assim uma maior flexibilidade de desenvolvimento.

Figura 38 - Gerando o APK.

Fonte: Autoria própria.

5 RESULTADOS E DISCUSSÕES

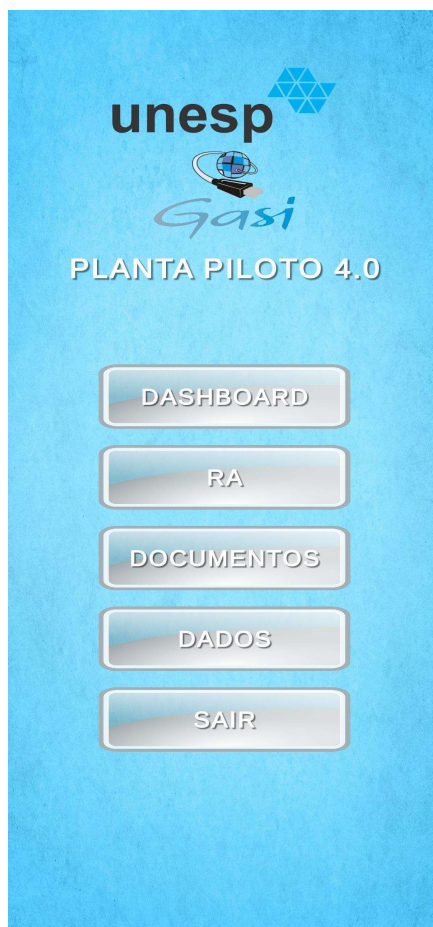
5.1 Aplicativo

Finalizado todo o desenvolvimento realizou-se os testes com o aplicativo rodando em um smartphone *Android* e os mesmos serão apresentados a seguir. O aplicativo desenvolvido é composto por cinco telas onde cada uma representa uma função diferente, um vídeo demonstrando o funcionamento do aplicativo pode ser encontrado em: https://youtu.be/jLTjhI_THAs.

5.1.1 Menu Inicial

O menu inicial está configurado para ser a primeira tela na abertura do aplicativo e através dessa tela é possível navegar pelas diferentes funcionalidades do aplicativo. A Figura 39 representa a tela inicial do aplicativo durante os testes.

Figura 39 - Menu Inicial.



Fonte: Acervo pessoal.

5.1.2 Dashboard

A primeira funcionalidade contida no menu inicial refere-se a abertura do dashboard de dados e permite que o usuário tenha as medições em tempo real de cada sensor sem a necessidade de estar próximo à bancada. Ao entrar na funcionalidade o usuário depara-se com a abertura do dashboard online, em que o mesmo contém um conjunto de medidores gauges acompanhados das suas respectivas unidades de medidas e valores em tempo real dos sensores. Também é possível, através de um menu, seleccionar-se as variáveis de forma independente para que se consulte o histograma da variável. A Figura 40 traz a funcionalidade rodando em um smartphone *Android*.

Figura 40 - Dashboard.



Fonte: Acervo pessoal.

5.1.3 Realidade Aumentada

A segunda funcionalidade exposta no menu inicial representa a abertura da realidade aumentada. Ao abrir a funcionalidade, o usuário depara-se com uma tela sem dados, apenas com a câmera aberta, ao passar a câmera pela bancada o *target* é reconhecido e inicia-se o processo de AR e torna-se possível o monitoramento do processo como mostra a Figura 41.

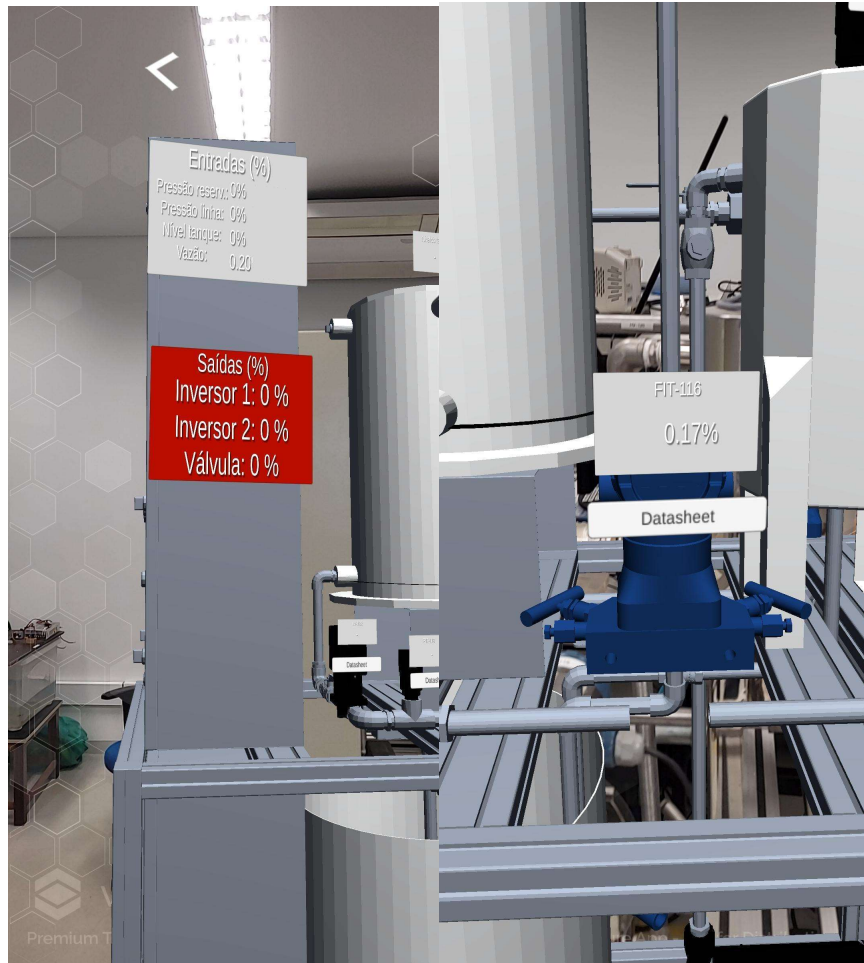
Figura 41 - Realidade Aumentada.



Fonte:Acervo pessoal.

Nota-se que inicialmente os dados dos sensores e atuadores estão divididos como entradas (sensores) e saídas (atuadores) e esses dados são os primeiros a serem observados, contudo ao aproximar-se dos sensores nota-se que cada um deles está com o respectivo valor medido e também com a opção de abertura do datasheet como mostra a Figura 42.

Figura 42 - Visualização dos dados na Realidade Aumentada.

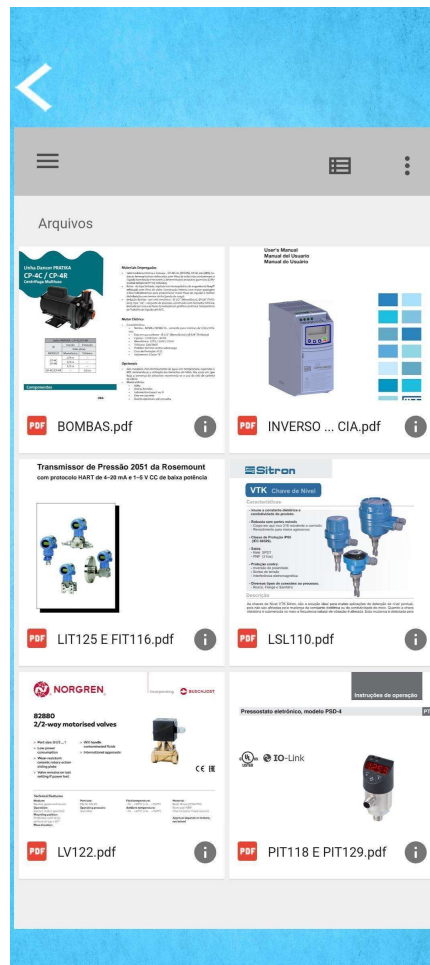


Fonte: Acervo pessoal.

5.1.4 Documentação

Outra funcionalidade desenvolvida foi para a reunião de toda a documentação referente a bancada de processos em um só lugar e assim facilitar na organização e procura pelos mesmos. Ao realizar-se o acesso a funcionalidade o usuário depara-se com uma tela que carrega informações do Google Drive e disponibiliza o acesso diretamente do aplicativo, através dela é possível realizar a leitura dos documentos e caso necessário fazer o download dos mesmos. Na Figura 43 é possível observar a funcionalidade desenvolvida.

Figura 43 - Documentos.



Fonte: Acervo pessoal.

5.1.5 Assistente Virtual para Consulta de Dados

Por fim, a última funcionalidade desenvolvida foi a possibilidade da consulta de dados referentes a bancada por dentro do aplicativo. A consulta é feita através do PLN onde o usuário digita sua pergunta e a mesma é interpretada e respondida pelo servidor, desta forma é possível consultar diversos dados referentes ao processo sem a necessidade de sair do aplicativo. Ao entrar na funcionalidade o usuário encontra a página onde é possível digitar sua pergunta e obter a resposta na tela. A Figura 44 representa a funcionalidade desenvolvida.

Figura 44 - Consulta de dados.



Fonte: Acervo pessoal.

5.2 Considerações Finais

Realizando-se os testes, nota-se que as funcionalidades desenvolvidas com o auxílio do UniWebView apresentam um funcionamento correto e mostra-se uma alternativa eficaz para a abertura de páginas web sem a necessidade da utilização do aplicativo do browser, além de permitir uma navegação fluida por todas as telas.

Durante os testes foi possível notar a grande facilidade na utilização da AR e como a mesma oferece uma vantagem no monitoramento do processo, além de facilitar a visualização dos dados. Um outro ponto notado foi que a separação dos dados com seu respectivo sensor e a inclusão do datasheet facilitam na tomada de decisão e proporcionam assim uma vantagem competitiva em comparação ao processo original.

Outra afirmação possível é de que a combinação entre o *Unity* e *Vuforia* se tornou adequada para a implementação, pois forneceu os recursos necessários para o desenvolvimento do trabalho e mostrou-se uma ótima alternativa em relação ao custo do desenvolvimento, visto que

ambos os softwares são gratuitos. Por fim a comunicação com o software *Molecular* para a requisição dos dados também se deu de forma satisfatória, conseguindo apresentar em tempo real os dados oriundos dos sensores da bancada de processos.

Deixa-se como sugestão futura a melhoria no processo de reconhecimento da AR, que está em constante evolução e em um futuro próximo pode permitir a utilização de novos recursos, a implementação da visão das malhas de controle e também de possíveis animações do processo com auxílio da Realidade Aumentada. Uma outra sugestão seria o desenvolvimento de novas funcionalidades ao aplicativo, como por exemplo o controle da planta.

6 CONCLUSÃO

O conceito de Indústria 4.0 para o monitoramento e controle de processos é uma poderosa ferramenta de gestão para as linhas produtivas das empresas, desta forma, este trabalho utiliza conhecimentos multidisciplinares que serão cada vez mais necessários para que um profissional possa desenvolver soluções inteligentes da Indústria 4.0.

O desenvolvimento de um aplicativo para monitoramento de processos industriais com auxílio de realidade aumentada além da possibilidade de, pelo mesmo aplicativo, consultar toda a documentação referente a planta, a visualização dos dados através do dashboard e a oportunidade de consulta a base de dados, através de uma assistente virtual, é um resultado satisfatório e mostra que os objetivos propostos foram alcançados.

O presente trabalho explorou os conceitos chaves de Indústria 4.0 juntamente com a aplicabilidade da realidade aumentada inserida neste contexto, além de seguir uma metodologia de desenvolvimento iterativo e uma arquitetura baseada em componentes. Para o desenvolvimento fundamentou-se quais os softwares que seriam utilizados, mostrando os benefícios do uso dos mesmos, e através da utilização de imagens representando os diagramas e esquemáticos foi possível uma melhor organização do conteúdo e a apresentação de ideias.

Todo o desenvolvimento trouxe os códigos e configurações realizadas bem como as ideias por trás de cada funcionalidade, de modo a servir de base para trabalhos futuros e com o foco em um maior entendimento por parte do leitor. Por fim, os resultados e considerações apresentados demonstram o sucesso deste trabalho e apontam provocações para trabalhos futuros.

O trabalho se mostrou de grande importância para o desenvolvimento científico devido a pouca exploração da Realidade Aumentada na indústria, além de gerar uma aplicação que, a partir dos conceitos apresentados, poderá ser estendida para outros trabalhos que utilizam Realidade Aumentada ou aplicativos móveis como foco da aplicação.

REFERÊNCIAS

- ANPEI. **O que é a Indústria 4.0?**. 2019. Disponível em: <<https://anpei.org.br/industria-4-0-o-que-e/>>. Acesso em: 5 jan. 2022.
- AUKSTAKALNIS, S. **Practical Augmented Reality: A guide to the technologies, applications, and human factors for AR and VR**. 2017, Addison-Wesley Professional.
- AUTOMATION WORD. **How Augmented Reality Became a Serious Tool for Manufacturing**. 2021. Disponível em: <<https://www.automationworld.com/process/iiot/article/21259479/how-augmented-reality-became-a-serious-tool-for-manufacturing>>. Acesso em: 5 jan. 2022.
- AZUMA, R. T. **A Survey of Augmented Reality**. 1997, Presence: Teleoperators & Virtual Environments, 6(4), pp 355-385
- BAENA, F.; GUARIN, A.; MORA, J.; SAUZA, J.; RETAT, S. **Learning Factory: The Path to Industry 4.0**, Procedia Manufacturing 9 (2017) 73–80, <https://doi.org/10.1016/j.promfg.2017.04.022>.
- BAJČI, B.; DUDIĆ, S.; ŠULC, J.; RELJIĆ, V.; ŠEŠLIJA, D. **Demonstration: Using remotely controlled one-way flow control valve for speed regulation of pneumatic cylinder**. In Smart Industry & Smart Education. REV 2018. Lecture Notes in Networks and Systems; Auer, M., Langmann, R., Eds.; Springer: Cham, Switzerland, 2018; ISBN 978-3-319-95677-0.
- BERG, L.; VANCE, J. **Industry use of virtual reality in product design and manufacturing: A survey**. 2017, Virtual Reality, 21(1), pp 1-17
- BIGDATACORP. **A Importância de uma Boa Visualização de Dados**. 2019. Disponível em: <<https://bigdatacorp.com.br/a-importancia-de-uma-boja-visualizacao-de-dados/>>. Acessado em: 05 jan.. 2022.
- BIGHETI, J. A.; RISSO, S. L.; CALDIERI, M. R.; GODOY, E. P. **Proposta de Arquitetura Orientada a Microserviços para Aplicações de Internet das Coisas Industrial**. In: XXII Congresso Brasileiro de Automática, 2018, João Pessoa. Anais do CBA 2018. Campinas: SBA, 2018
- BLOEM, J.; VAN DOORN, M.; DUIVESTEN, S.; EXCOFFIER, D.; MASS, R.; VAN OMMEREN, E. **The Fourth Industrial Revolution—Things to Tighten the Link between IT and OT**. Sogeti VINT. 2014. Available online: <https://www.sogeti.com/globalassets/global/special/sogeti-things3en.pdf> (accessed on 9 April 2021).
- BLOG IMPACTA. **Você sabe o que é Visual Studio?**. 2017. Disponível em: <<https://www.impacta.com.br/blog/voce-sabe-o-que-e-visual-studio/>>. Acesso em: 5 jan. 2022.
- CARMIGNIANI, J.; FURHT, B.; ANISETTI, M.; CERAVOLO, P.; DAMIANI, E.; IVKOVIC, M. **Augmented reality technologies, systems and applications**. Multimed. Tools Appl. 2011, 51, 341–377.
- DEVELOPERS; **WebView**. 2022. Disponível em: <<https://developer.android.com/reference/android/webkit/WebView>>. Acesso em: 5 jan. 2022.

DINI, G.; DALLE MURA, M. **Application of augmented reality techniques in through-life engineering services**. Procedia CIRP 2015, 38, 14–23.

DOMINGUES, F. O. **Historiador de processos como um microsserviço: uma abordagem para a Indústria 4.0**. 2021. 89 f. Trabalho de Graduação (Bacharel em Engenharia de Controle e Automação) - Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista, Sorocaba, 2021.

FAST-BERGLUND, Å.; GONG, L.; LI, D. **Testing and validating Extended Reality (xR) technologies in manufacturing**. 2018, Procedia Manufacturing, 25, pp 31-38

FERNANDEZ, D. P. **Assistente Virtual baseada em Processamento de Linguagem Natural para Automação de Consulta em Banco de Dados Industrial**. 2021. 89 f. Trabalho de Graduação (Bacharel em Engenharia de Controle e Automação) - Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista, Sorocaba, 2021.

FRAGA-LAMAS, P.; FERNÁNDEZ-CARAMÉS, T.M.; BLANCO-NOVOA, O.; VILAR-MONTESINOS, M.A. **A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard**, IEEE Access 6 (2018) 13358–13375. 10.1109/ACCESS.2018.2808326.

FRÈRE, E.; ZURECK, A.; RÖHRIG, K. **“Industry 4.0 in Germany - The Obstacles Regarding Smart Production in the Manufacturing Industry,”** SSRN Electron. J., Jan. 2018, doi: 10.2139/ssrn.3223765

GRIECO, A.; CARICATO, P.; GIANFREDI, D.; PESCE, M.; RIGON, V.; TREGNAGHI, L.; VOGLINO, A. **An Industry 4.0 Case Study in Fashion Manufacturing**, Procedia Manuf. 11 (2017) 871-877, <https://doi.org/10.1016/j.promfg.2017.07.190>.

HOŘEJŠÍ, P. **Augmented Reality System for Virtual Training of Parts Assembly**, Procedia Eng. 100 (2015) 699–706, <https://doi.org/10.1016/j.proeng.2015.01.422>.

INTERSOG. **Why Unity 3D Is So Popular for Games and VR Development**. 2020. Disponível em: <<https://intersog.com/blog/why-vr-and-games-made-with-unity-are-so-impressive/>>. Acesso em: 5 jan. 2022.

LEYH, C.; MARTIN, S.; SCHÄFFER, T. **Industry 4.0 and Lean Production – A Matching Relationship? An analysis of selected Industry 4.0 models**, 2017 Federal Conference on Computer Science and Informatics Systems (FedCSIS) Prague 11 (2017) 989-993, <https://doi.org/10.15439/2017F365>.

LINKEDIN. **5 reasons why Unity is one of the best game engines for beginners**. 2019. Disponível em: <<https://www.linkedin.com/pulse/5-reasons-why-unity-one-best-game-engines-beginners-ghenea/>>. Acesso em: 5 jan. 2022.

LU, Y. **Industry 4.0: A survey on technologies, applications and open research issues**, J. Ind. Information Integr. 6 (2017) 1–10. 10.1016/j.jii.2017.04.005.

MASOOD, T.; EGGER, J. **Adopting augmented reality in the age of industrial digitalisation**. Comput. Ind. 2020, 115.

MICROSOFT. **Visual Studio: IDE e Editor de Código para Desenvolvedores de Software e Teams**. 2022. Disponível em: <<https://visualstudio.microsoft.com/pt-br/>>. Acesso em: 5 jan. 2022.

MILGRAM, P.; TAKEMURA, H.; UTSUMI, A.; KISHINO, F. **Augmented reality: A class of displays on the reality-virtuality continuum**. Telemanip. Telepresence Technol. 1995, 2351, 282–292

MOLECULER. Fast & powerful microservices framework for Node.js. Disponível em: <https://moleculer.services/>. Acesso em: 28 jan. 2022.

MOLECULER. **O que é o Moleculer?**. 2022. Disponível em: <<https://moleculer.services/pt/docs/0.14/index.html>>. Acesso em: 5 jan. 2022.

MOORE, G. E. **Cramming More Components onto Integrated Circuits**. 1998, Proceedings of the IEEE, 86(1), pp 82-85

MOORE, G. E. **Lithography and the future of Moore's law**. 1995, Paper presented at the Integrated Circuit Metrology, Inspection, and Process Control IX, 2439 pp 2-17

MOTYL, B.; BARONIO, G.; UBERTI, S.; SPERANZA, D.; FILIPPI, S. **How will Change the Future Engineer's Skills in the Industry 4.0 Framework? A questionnaire Survey**, Procedia Manuf. 11 (2017) 1501–1509, <https://doi.org/10.1016/j.promfg.2017.07.282>.

PERUZZINI, M.; GRANDI, F.; PELLICCIARI, M. Benchmarking of Tools for User **Experience Analysis in Industry 4.0**, Procedia Manuf. 11 (2017), 806–813, <https://doi.org/10.1016/j.promfg.2017.07.182>.

PONTAROLLI, R. P. **Composição de serviços e mecanismos de segurança para arquiteturas orientadas a microserviços**. Dissertação (mestrado). Universidade Estadual Paulista. Instituto de Ciência e Tecnologia de Sorocaba. Sorocaba. 2020.

PORTAL DA INDUSTRIA. **O que é a indústria 4.0?**. 2022. Disponível em: <<https://www.portaldaindustria.com.br/industria-de-a-z/industria-4-0/>>. Acesso em: 5 jan. 2022.

PROGRAMMING HISTORIAN. **Creating Mobile Augmented Reality Experiences in Unity**. 2021. Disponível em: <<https://programminghistorian.org/en/lessons/creating-mobile-augmented-reality-experiences-in-unity>>. Acesso em: 5 jan. 2022.

PTC. **Vuforia: Market-Leading Enterprise AR**. 2022. Disponível em: <<https://www.ptc.com/en/products/vuforia>>. Acesso em: 5 jan. 2022.

RAYWENDERLICH. **Introduction to TextMesh Pro in Unity**. 2021. Disponível em: <<https://www.raywenderlich.com/22175776-introduction-to-textmesh-pro-in-unity>>. Acesso em: 5 jan. 2022.

RUNJI, J. M.; LIN, C. Y. **“Markerless cooperative augmented reality-based smart manufacturing double-check system: Case of safe PCBA inspection following automatic optical inspection.”** Robotics and Computer-Integrated Manufacturing, v. 64, n. February, p. 101957, 2020.

SALENIUS, M. *Designing A Data Solution For A Game Company – Case InfluxDB and Grafana*. 2017. Disponível em: <<https://www.theseus.fi/handle/10024/137958>>. Acessado em: 05 jan. 2022

SALKIN, C.; ONER, M.; USTUNDAG, A.; CEVIKCAN, E. **A Conceptual Framework for Industry 4.0**, In *Industry 4.0: Managing the Digital Transformation*, Springer Series in Advanced Manufacturing, Springer, Cham (2018) 3–23. 10.1007/978-3-319-57870-5.

SHERMAN, W.R.; CRAIG, A.B. **Understanding virtual reality: Interface, application, and design**, 2019 2nd edn. Morgan Kaufmann, San Francisco, CA

SLIDE FACTORY. **Using UnityWebRequest to Download Resources in Unity**. 2020. Disponível em: <<https://theslidefactory.com/using-unitywebrequest-to-download-resources-in-unity/>>. Acesso em: 5 out. 2021.

SREEKANTA, M. H.; SARODE, A.; GEORGE, K. **“Error Detection using Augmented Reality in the Subtractive Manufacturing Process”**. 2020 10th Annual Computing and Communication Workshop and Conference, CCWC 2020, p. 592–597, 2020.

SYBERFELDT, A.; DANIELSSON, O.; GUSTAVSSON, P. **Augmented Reality Smart Glasses in the Smart Factory: Product Evaluation Guidelines and Review of Available Products**, IEEE Access 5 (2017) 9118–9130, <https://doi.org/10.1109/ACCESS.2017.2703952>.

SYBERFELDT, A.; DANIELSSON, O.; HOLM, M.; WANG, L. **Visual Assembling Guidance Using Augmented Reality**, Procedia Manuf. 1 (2015) 98–109, <https://doi.org/10.1016/j.promfg.2015.09.068>.

SYBERFELDT, A.; DANIELSSON, O.; HOLM, M.; WANG, L.; BREWSTER, R.L. **Brewster, Support Systems on the Industrial Shop-floor of the Future – Operator’s Perspective on Augmented Reality**, Procedia CIRP 44 (2016) 108–113, <https://doi.org/10.1016/j.procir.2016.02.017>.

SYBERFELDT, A.; HOLM, M.; DANIELSSON, O.; WANG, L.; BREWSTER, R.L. **Support systems on the industrial shop-floors of the future—operators’ perspective on augmented reality**. Procedia CIRP 2016, 44, 108–113.

TOSHNIWAL, R.; DASTIDAR, K. G. **Virtual reality: The future interface of technology**. International Journal of Computer Science and Information Technologies, v. 5, p. 7032–7034, 2014.

TUTORIALS FOR AR. **Day 6 : How to use Vuforia in Unity to create AR Application**. 2019. Disponível em: <<https://tutorialsforar.com/how-to-use-vuforia-in-unity-to-create-ar-application/>>. Acesso em: 5 jan. 2022.

UNITY TECHNOLOGIES. **Coroutines**. 2022. Disponível em: <<https://docs.unity3d.com/Manual/Coroutines.html>>. Acesso em: 5 out. 2021.

UNITY TECHNOLOGIES. **TextMeshPro**. 2022. Disponível em: <<https://docs.unity3d.com/Manual/com.unity.textmeshpro.html>>. Acesso em: 5 jan. 2022.

UNITY TECHNOLOGIES. **UnityWebRequest**. 2022. Disponível em: <<https://docs.huiohoo.com/unity/5.4/Documentation/en/Manual/UnityWebRequest.html>>. Acesso em: 5 out. 2021.

UNIWEBVIEW. **UniWebView**. 2022. Disponível em: <<https://docs.uniwebview.com/>>. Acesso em: 5 jan. 2022.

V2COM. **Indústria 4.0: dados atualizados sobre o Brasil**. 2021. Disponível em: <<https://v2com.com/2021/03/10/industria-4-0/>>. Acesso em: 5 jul. 2022.

VUFORIA. **Model Target Generator User Guide**. 2022. Disponível em: <<https://library.vuforia.com/objects/model-target-generator-user-guide>>. Acesso em: 5 out. 2021.

WAGNER, T.; HERRMANN, C.; THIEDE, S. **Industry 4.0 Impacts on Lean Production Systems**, Procedia CIRP 63 (2017) 125–131. 10.1016/j.procir.2017.02.041.

WANG, P.; WU, P.; WANG, J.; CHI, H.; WANG, X. **A Critical Review of the Use of Virtual Reality in Construction Engineering Education and Training**. 2018, International Journal of Environmental Research and Public Health, 15(6).

WEYER, S.; SCHMITT, M.; OHMER, M.; GORECKY, D. **Towards Industry 4.0 – Standardization as the crucial challenge for highly modular, multi-vendor production systems**, IFAC-PapersOnLine 48 (3) (2015) 579-584, <https://doi.org/10.1016/j.ifacol.2015.06.143>.

WHYTE, J.; NIKOLIĆ, D. **Virtual Reality and the Built Environment**. 2018, Routledge, London