

Master's Degree Artificial Intelligence in Programming



Master's Degree Artificial Intelligence in Programming

- » Modality: online
- » Duration: 12 months
- » Certificate: TECH Global University
- » Credits: 90 ECTS
- » Schedule: at your own pace
- » Exams: online

Website: www.techtitute.com/us/artificial-intelligence/master-degree/master-artificial-intelligence-programming

Index

01

Introduction to the Program

p. 4

02

Why Study at TECH?

p. 8

03

Syllabus

p. 12

04

Teaching Objectives

p. 32

05

Career Opportunities

p. 42

06

Software Licenses Included

p. 46

07

Study Methodology

p. 50

08

Teaching Staff

p. 60

09

Certificate

p. 64

01

Introduction to the Program

In an era characterized by technological advances, Artificial Intelligence (AI) has emerged as a primary tool for programming experts. Its importance lies in its ability to mechanize arduous activities, make decisions based on accurate data and learn from patterns. Therefore, Machine Learning offers valuable techniques for computer scientists to design more intelligent systems. These range from algorithms to provide programs with greater thoroughness to the development of autonomous systems, useful for changing the way code is executed. For this reason, TECH has developed a comprehensive university degree that will provide students with the latest advances in this field. All under a 100% online methodology, adapted to the agenda of busy professionals.



“

With this completely online Master's Degree, you will design personalized and intuitive user experiences to optimize customer satisfaction”

Computational Intelligence helps institutions improve productivity in software development. Its tools have the capacity to handle unstructured data, learn from past experiences and adapt to changes in dynamic environments. In addition, intelligent systems can predict potential application problems before they happen, allowing professionals to take preventive measures to avoid costly problems in the future. In this context, the most prestigious international IT companies are actively seeking to incorporate specialists in software architecture for QA testing.

In this context, TECH is implementing an innovative Master's Degree in Artificial Intelligence in Programming. Designed by top experts, the syllabus will delve into the training of algorithms to develop products with intelligent systems. Likewise, the syllabus will delve into the essential extensions for Visual Studio Code, the most widely used source code editor today. On the other hand, the teaching materials will address the integration of Artificial Intelligence in management with databases to detect possible failures and create unit tests. In this way, students will gain the technical skills to develop advanced solutions based on machine learning techniques, optimizing the performance and efficiency of programming processes.

It should be noted that this university program is taught using a completely online methodology. In this way, professionals will be able to balance their work with their knowledge updating process. In addition, they will enjoy the support of a highly specialized teaching staff that will provide them with personalized guidance. In this sense, the only requirement for entering the Virtual Campus is that students have an electronic device with an Internet connection, being able to connect even from their cell phone.

This **Master's Degree in Artificial Intelligence in Programming** contains the most complete and up-to-date program on the market. The most important features include:

- ♦ Development of practical cases presented by experts in Artificial Intelligence in Programming
- ♦ The graphic, schematic, and practical contents with which they are created, provide scientific and practical information on the disciplines that are essential for professional practice
- ♦ Practical exercises where the self-assessment process can be carried out to improve learning
- ♦ Special emphasis on innovative methodologies in Artificial Intelligence in Programming
- ♦ Theoretical lessons, questions to the expert, debate forums on controversial topics, and individual reflection assignments
- ♦ Content that is accessible from any fixed or portable device with an internet connection



You will implement advanced Natural Language Processing techniques for the development of virtual assistants such as chatbots"

“

You will delve into the optimization of the performance of Artificial Intelligence models by applying methods of adjustment, validation and deployment in productive environments"

Its teaching staff includes professionals from the field of Artificial Intelligence in Programming, who bring their work experience to this program, as well as renowned specialists from leading companies and prestigious universities.

The multimedia content, developed with the latest educational technology, will provide the professional with situated and contextual learning, i.e., a simulated environment that will provide an immersive learning experience designed to prepare for real-life situations.

This program is designed around Problem-Based Learning, whereby the student must try to solve the different professional practice situations that arise throughout the program. For this purpose, the professional will be assisted by an innovative interactive video system created by renowned and experienced experts.

You will have a solid understanding of the testing life cycle, from the creation of test cases to the detection of bugs.

TECH's Relearning method will allow you to learn with less effort and higher performance, involving you more in your professional specialization.



02

Why Study at TECH?

TECH is the world's largest online university. With an impressive catalog of more than 14,000 university programs, available in 11 languages, it is positioned as a leader in employability, with a 99% job placement rate. In addition, it has a huge faculty of more than 6,000 professors of the highest international prestige.



“

Study at the largest online university in the world and ensure your professional success. The future begins at TECH”

The world's best online university, according to FORBES

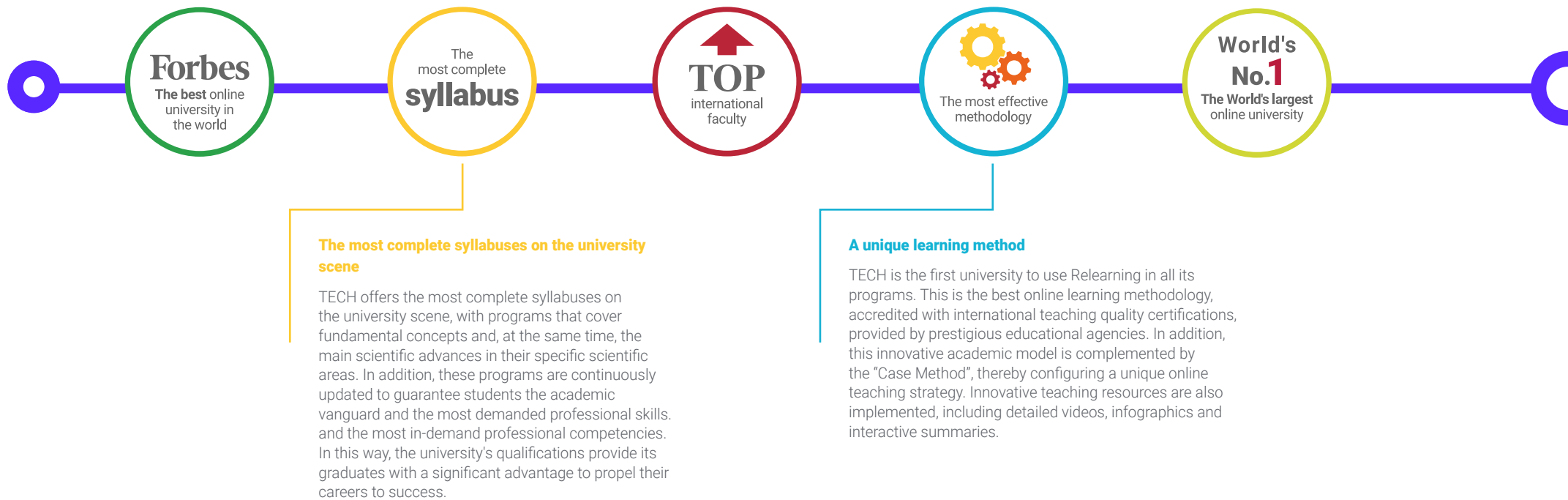
The prestigious Forbes magazine, specialized in business and finance, has highlighted TECH as "the best online university in the world" This is what they have recently stated in an article in their digital edition in which they echo the success story of this institution, "thanks to the academic offer it provides, the selection of its teaching staff, and an innovative learning method oriented to form the professionals of the future".

The best top international faculty

TECH's faculty is made up of more than 6,000 professors of the highest international prestige. Professors, researchers and top executives of multinational companies, including Isaiah Covington, performance coach of the Boston Celtics; Magda Romanska, principal investigator at Harvard MetaLAB; Ignacio Wistuba, chairman of the department of translational molecular pathology at MD Anderson Cancer Center; and D.W. Pine, creative director of TIME magazine, among others.

The world's largest online university

TECH is the world's largest online university. We are the largest educational institution, with the best and widest digital educational catalog, one hundred percent online and covering most areas of knowledge. We offer the largest selection of our own degrees and accredited online undergraduate and postgraduate degrees. In total, more than 14,000 university programs, in ten different languages, making us the largest educational institution in the world.



The official online university of the NBA

TECH is the official online university of the NBA. Thanks to our agreement with the biggest league in basketball, we offer our students exclusive university programs, as well as a wide variety of educational resources focused on the business of the league and other areas of the sports industry. Each program is made up of a uniquely designed syllabus and features exceptional guest hosts: professionals with a distinguished sports background who will offer their expertise on the most relevant topics.

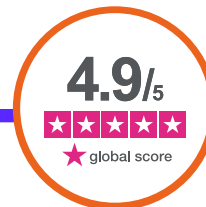
Leaders in employability

TECH has become the leading university in employability. Ninety-nine percent of its students obtain jobs in the academic field they have studied within one year of completing any of the university's programs. A similar number achieve immediate career enhancement. All this thanks to a study methodology that bases its effectiveness on the acquisition of practical skills, which are absolutely necessary for professional development.



Google Premier Partner

The American technology giant has awarded TECH the Google Premier Partner badge. This award, which is only available to 3% of the world's companies, highlights the efficient, flexible and tailored experience that this university provides to students. The recognition not only accredits the maximum rigor, performance and investment in TECH's digital infrastructures, but also places this university as one of the world's leading technology companies.



The top-rated university by its students

Students have positioned TECH as the world's top-rated university on the main review websites, with a highest rating of 4.9 out of 5, obtained from more than 1,000 reviews. These results consolidate TECH as the benchmark university institution at an international level, reflecting the excellence and positive impact of its educational model.



03 Syllabus

This Master's Degree will provide graduates with a holistic approach, which will give them a significant advantage in computer development, equipping them with specific skills. To achieve this, the syllabus will cover everything from preparing the development environment to optimizing software and implementing intelligent systems in real projects. The syllabus will also delve into aspects such as no-code interface design, using ChatGPT to optimize code, and the application of machine learning in QA testing.



“

You will implement Artificial Intelligence solutions in Cloud and Edge Computing environments, ensuring accessibility to all resources"

Module 1. Fundamentals of Artificial Intelligence

- 1.1. History of Artificial Intelligence
 - 1.1.1. When Do We Start Talking About Artificial Intelligence?
 - 1.1.2. References in Film
 - 1.1.3. Importance of Artificial Intelligence
 - 1.1.4. Technologies that Enable and Support Artificial Intelligence
- 1.2. Artificial Intelligence in Games
 - 1.2.1. Game Theory
 - 1.2.2. Minimax and Alpha-Beta Pruning
 - 1.2.3. Simulation: Monte Carlo
- 1.3. Neural Networks
 - 1.3.1. Biological Fundamentals
 - 1.3.2. Computational Model
 - 1.3.3. Supervised and Unsupervised Neural Networks
 - 1.3.4. Simple Perceptron
 - 1.3.5. Multilayer Perceptron
- 1.4. Genetic Algorithms
 - 1.4.1. History
 - 1.4.2. Biological Basis
 - 1.4.3. Problem Coding
 - 1.4.4. Generation of the Initial Population
 - 1.4.5. Main Algorithm and Genetic Operators
 - 1.4.6. Evaluation of Individuals: Fitness
- 1.5. Thesauri, Vocabularies, Taxonomies
 - 1.5.1. Vocabulary
 - 1.5.2. Taxonomy
 - 1.5.3. Thesauri
 - 1.5.4. Ontologies
 - 1.5.5. Knowledge Representation: Semantic Web
- 1.6. Semantic Web
 - 1.6.1. Specifications: RDF, RDFS and OWL
 - 1.6.2. Inference/ Reasoning
 - 1.6.3. *Linked Data*

- 1.7. Expert Systems and DSS
 - 1.7.1. Expert Systems
 - 1.7.2. Decision Support Systems
- 1.8. Chatbots and Virtual Assistants
 - 1.8.1. Types of Assistants: Voice and Text Assistants
 - 1.8.2. Fundamental Parts for the Development of an Assistant: Intents, Entities and Dialog Flow
 - 1.8.3. Integrations: Web, Slack, WhatsApp, Facebook
 - 1.8.4. Assistant Development Tools: Dialog Flow, Watson Assistant
- 1.9. AI Implementation Strategy
- 1.10. Future of Artificial Intelligence
 - 1.10.1. Understand How to Detect Emotions Using Algorithms
 - 1.10.2. Creating a Personality: Language, Expressions and Content
 - 1.10.3. Trends of Artificial Intelligence
 - 1.10.4. Reflections

Module 2. Data Types and Life Cycle

- 2.1. Statistics
 - 2.1.1. Statistics: Descriptive Statistics, Statistical Inferences
 - 2.1.2. Population, Sample, Individual
 - 2.1.3. Variables: Definition, Measurement Scales
- 2.2. Types of Statistical Data
 - 2.2.1. By Type
 - 2.2.1.1. Quantitative: Continuous Data and Discrete Data
 - 2.2.1.2. Qualitative: Binomial Data, Nominal Data and Ordinal Data
 - 2.2.2. By Form
 - 2.2.2.1. Numerical
 - 2.2.2.2. Text
 - 2.2.2.3. Logical
 - 2.2.3. By Source
 - 2.2.3.1. Primary
 - 2.2.3.2. Secondary

- 2.3. Data Lifecycle
 - 2.3.1. Lifecycle Stages
 - 2.3.2. Lifecycle Milestones
 - 2.3.3. FAIR Principles
- 2.4. Initial Stages of the Cycle
 - 2.4.1. Goal Definition
 - 2.4.2. Determination of Required Resources
 - 2.4.3. Gantt Chart
 - 2.4.4. Data Structure
- 2.5. Data Collection
 - 2.5.1. Data Collection Methodology
 - 2.5.2. Data Collection Tools
 - 2.5.3. Data Collection Channels
- 2.6. Data Cleaning
 - 2.6.1. Data Cleaning Phases
 - 2.6.2. Data Quality
 - 2.6.3. Data Manipulation (using R)
- 2.7. Data Analysis, Interpretation and Evaluation of Results
 - 2.7.1. Statistical Measures
 - 2.7.2. Relationship Indices
 - 2.7.3. Data Mining
- 2.8. Data Warehouse
 - 2.8.1. Components of a Data Warehouse
 - 2.8.2. Design
 - 2.8.3. Aspects to Consider
- 2.9. Data Availability
 - 2.9.1. Utility
 - 2.9.2. Security
- 2.10. Regulatory Framework
 - 2.10.1. Good Practices

Module 3. Data in Artificial Intelligence

- 3.1. Data Science
 - 3.1.1. Data Science
 - 3.1.2. Advanced Tools for Data Scientists
- 3.2. Data, Information and Knowledge
 - 3.2.1. Data, Information and Knowledge
 - 3.2.2. Types of Data
 - 3.2.3. Data Sources
- 3.3. From Data to Information
 - 3.3.1. Data Analysis
 - 3.3.2. Types of Analysis
 - 3.3.3. Extraction of Information from a Dataset
- 3.4. Extraction of Information Through Visualization
 - 3.4.1. Visualization as an Analysis Tool
 - 3.4.2. Visualization Methods
 - 3.4.3. Visualization of a Data Set
- 3.5. Data Quality
 - 3.5.1. Quality Data
 - 3.5.2. Data Cleaning
 - 3.5.3. Basic Data Pre-Processing
- 3.6. *Dataset*
 - 3.6.1. Dataset Enrichment
 - 3.6.2. The Curse of Dimensionality
 - 3.6.3. Modification of Our Data Set
- 3.7. Unbalance
 - 3.7.1. Classes of Unbalance
 - 3.7.2. Unbalance Mitigation Techniques
 - 3.7.3. Balancing a Dataset
- 3.8. Unsupervised Models
 - 3.8.1. Unsupervised Model
 - 3.8.2. Methods
 - 3.8.3. Classification with Unsupervised Models

- 3.9. Supervised Models
 - 3.9.1. Supervised Model
 - 3.9.2. Methods
 - 3.9.3. Classification with Supervised Models
- 3.10. Tools and Good Practices
 - 3.10.1. Good Practices for Data Scientists
 - 3.10.2. The Best Model
 - 3.10.3. Useful Tools

Module 4. Data Mining. Selection, Pre-Processing and Transformation

- 4.1. Statistical Inference
 - 4.1.1. Descriptive Statistics vs. Statistical Inference
 - 4.1.2. Parametric Procedures
 - 4.1.3. Non-Parametric Procedures
- 4.2. Exploratory Analysis
 - 4.2.1. Descriptive Analysis
 - 4.2.2. Visualization
 - 4.2.3. Data Preparation
- 4.3. Data Preparation
 - 4.3.1. Integration and Data Cleaning
 - 4.3.2. Normalization of Data
 - 4.3.3. Transforming Attributes
- 4.4. Missing Values
 - 4.4.1. Treatment of Missing Values
 - 4.4.2. Maximum Likelihood Imputation Methods
 - 4.4.3. Missing Value Imputation Using Machine Learning
- 4.5. Noise in the Data
 - 4.5.1. Noise Classes and Attributes
 - 4.5.2. Noise Filtering
 - 4.5.3. The Effect of Noise
- 4.6. The Curse of Dimensionality
 - 4.6.1. *Oversampling*
 - 4.6.2. *Undersampling*
 - 4.6.3. Multidimensional Data Reduction

- 4.7. From Continuous to Discrete Attributes
 - 4.7.1. Continuous Data vs. Discrete Data
 - 4.7.2. Discretization Process
- 4.8. The Data
 - 4.8.1. Data Selection
 - 4.8.2. Prospects and Selection Criteria
 - 4.8.3. Selection Methods
- 4.9. Instance Selection
 - 4.9.1. Methods for Instance Selection
 - 4.9.2. Prototype Selection
 - 4.9.3. Advanced Methods for Instance Selection
- 4.10. Data Pre-Processing in Big Data Environments

Module 5. Algorithm and Complexity in Artificial Intelligence

- 5.1. Introduction to Algorithm Design Strategies
 - 5.1.1. Recursion
 - 5.1.2. Divide and Conquer
 - 5.1.3. Other Strategies
- 5.2. Efficiency and Analysis of Algorithms
 - 5.2.1. Efficiency Measures
 - 5.2.2. Measuring the Size of the Input
 - 5.2.3. Measuring Execution Time
 - 5.2.4. Worst, Best and Average Case
 - 5.2.5. Asymptotic Notation
 - 5.2.6. Criteria for Mathematical Analysis of Non-Recursive Algorithms
 - 5.2.7. Mathematical Analysis of Recursive Algorithms
 - 5.2.8. Empirical Analysis of Algorithms
- 5.3. Sorting Algorithms
 - 5.3.1. Concept of Sorting
 - 5.3.2. Bubble Sorting
 - 5.3.3. Sorting by Selection
 - 5.3.4. Sorting by Insertion
 - 5.3.5. Sorting by Merge (Merge_Sort)
 - 5.3.6. Sorting Quickly (Quick_Sort)

- 5.4. Algorithms with Trees
 - 5.4.1. Tree Concept
 - 5.4.2. Binary Trees
 - 5.4.3. Tree Paths
 - 5.4.4. Representing Expressions
 - 5.4.5. Ordered Binary Trees
 - 5.4.6. Balanced Binary Trees
- 5.5. Algorithms Using Heaps
 - 5.5.1. Heaps
 - 5.5.2. The Heapsort Algorithm
 - 5.5.3. Priority Queues
- 5.6. Graph Algorithms
 - 5.6.1. Representation
 - 5.6.2. Traversal in Width
 - 5.6.3. Depth Travel
 - 5.6.4. Topological Sorting
- 5.7. Greedy Algorithms
 - 5.7.1. Greedy Strategy
 - 5.7.2. Greedy Strategy Elements
 - 5.7.3. Currency Exchange
 - 5.7.4. Traveler's Problem
 - 5.7.5. Backpack Problem
- 5.8. Minimal Path Finding
 - 5.8.1. The Minimum Path Problem
 - 5.8.2. Negative Arcs and Cycles
 - 5.8.3. Dijkstra's Algorithm
- 5.9. Greedy Algorithms on Graphs
 - 5.9.1. Minimum Spanning Tree
 - 5.9.2. Prim's Algorithm
 - 5.9.3. Kruskal's Algorithm
 - 5.9.4. Complexity Analysis
- 5.10. *Backtracking*
 - 5.10.1. Backtracking Algorithm
 - 5.10.2. Alternative Techniques

Module 6. Intelligent Systems

- 6.1. Agent Theory
 - 6.1.1. Concept History
 - 6.1.2. Agent Definition
 - 6.1.3. Agents in Artificial Intelligence
 - 6.1.4. Agents in Software Engineering
- 6.2. Agent Architectures
 - 6.2.1. The Reasoning Process of an Agent
 - 6.2.2. Reactive Agents
 - 6.2.3. Deductive Agents
 - 6.2.4. Hybrid Agents
 - 6.2.5. Comparison
- 6.3. Information and Knowledge
 - 6.3.1. Difference between Data, Information and Knowledge
 - 6.3.2. Data Quality Assessment
 - 6.3.3. Data Collection Methods
 - 6.3.4. Information Acquisition Methods
 - 6.3.5. Knowledge Acquisition Methods
- 6.4. Knowledge Representation
 - 6.4.1. The Importance of Knowledge Representation
 - 6.4.2. Definition of Knowledge Representation According to Roles
 - 6.4.3. Knowledge Representation Features
- 6.5. Ontologies
 - 6.5.1. Introduction to Metadata
 - 6.5.2. Philosophical Concept of Ontology
 - 6.5.3. Computing Concept of Ontology
 - 6.5.4. Domain Ontologies and Higher-Level Ontologies
 - 6.5.5. How to Build an Ontology

6.6. Ontology Languages and Ontology Creation Software

6.6.1. Triple RDF, Turtle and N

6.6.2. RDF Schema

6.6.3. OWL

6.6.4. SPARQL

6.6.5. Introduction to Ontology Creation Tools

6.6.6. Installing and Using Protégé

6.7. Semantic Web

6.7.1. Current and Future Status of the Semantic Web

6.7.2. Semantic Web Applications

6.8. Other Knowledge Representation Models

6.8.1. Vocabulary

6.8.2. Global Vision

6.8.3. Taxonomy

6.8.4. Thesauri

6.8.5. Folksonomy

6.8.6. Comparison

6.8.7. Mind Maps

6.9. Knowledge Representation Assessment and Integration

6.9.1. Zero-Order Logic

6.9.2. First-Order Logic

6.9.3. Descriptive Logic

6.9.4. Relationship between Different Types of Logic

6.9.5. Prolog: Programming Based on First-Order Logic

6.10. Semantic Reasoners, Knowledge-Based Systems and Expert Systems

6.10.1. Concept of Reasoner

6.10.2. Reasoner Applications

6.10.3. Knowledge-Based Systems

6.10.4. MYCIN: History of Expert Systems

6.10.5. Expert Systems Elements and Architecture

6.10.6. Creating Expert Systems

Module 7. Machine Learning and Data Mining

7.1. Introduction to Knowledge Discovery Processes and Basic Concepts of Machine Learning

7.1.1. Key Concepts of Knowledge Discovery Processes

7.1.2. Historical Perspective of Knowledge Discovery Processes

7.1.3. Stages of the Knowledge Discovery Processes

7.1.4. Techniques Used in Knowledge Discovery Processes

7.1.5. Characteristics of Good Machine Learning Models

7.1.6. Types of Machine Learning Information

7.1.7. Basic Learning Concepts

7.1.8. Basic Concepts of Unsupervised Learning

7.2. Data Exploration and Pre-Processing

7.2.1. Data Processing

7.2.2. Data Processing in the Data Analysis Flow

7.2.3. Types of Data

7.2.4. Data Transformations

7.2.5. Visualization and Exploration of Continuous Variables

7.2.6. Visualization and Exploration of Categorical Variables

7.2.7. Correlation Measures

7.2.8. Most Common Graphic Representations

7.2.9. Introduction to Multivariate Analysis and Dimensionality Reduction

7.3. Decision Trees

7.3.1. ID Algorithm

7.3.2. Algorithm C

7.3.3. Overtraining and Pruning

7.3.4. Result Analysis

7.4. Evaluation of Classifiers

7.4.1. Confusion Matrices

7.4.2. Numerical Evaluation Matrices

7.4.3. Kappa Statistic

7.4.4. ROC Curves



- 7.5. Classification Rules
 - 7.5.1. Rule Evaluation Measures
 - 7.5.2. Introduction to Graphic Representation
 - 7.5.3. Sequential Overlay Algorithm
- 7.6. Neural Networks
 - 7.6.1. Basic Concepts
 - 7.6.2. Simple Neural Networks
 - 7.6.3. Backpropagation Algorithm
 - 7.6.4. Introduction to Recurrent Neural Networks
- 7.7. Bayesian Methods
 - 7.7.1. Basic Probability Concepts
 - 7.7.2. Bayes' Theorem
 - 7.7.3. Naive Bayes
 - 7.7.4. Introduction to Bayesian Networks
- 7.8. Regression and Continuous Response Models
 - 7.8.1. Simple Linear Regression
 - 7.8.2. Multiple Linear Regression
 - 7.8.3. Logistic Regression
 - 7.8.4. Regression Trees
 - 7.8.5. Introduction to Support Vector Machines (SVM)
 - 7.8.6. Goodness-of-Fit Measures
- 7.9. *Clustering*
 - 7.9.1. Basic Concepts
 - 7.9.2. Hierarchical Clustering
 - 7.9.3. Probabilistic Methods
 - 7.9.4. EM Algorithm
 - 7.9.5. B-Cubed Method
 - 7.9.6. Implicit Methods
- 7.10. Text Mining and Natural Language Processing (NLP)
 - 7.10.1. Basic Concepts
 - 7.10.2. Corpus Creation
 - 7.10.3. Descriptive Analysis
 - 7.10.4. Introduction to Feelings Analysis

Module 8. Neural Networks, the Basis of Deep Learning

- 8.1. Deep Learning
 - 8.1.1. Types of Deep Learning
 - 8.1.2. Applications of Deep Learning
 - 8.1.3. Advantages and Disadvantages of Deep Learning
- 8.2. Operations
 - 8.2.1. Sum
 - 8.2.2. Product
 - 8.2.3. Transfer
- 8.3. Layers
 - 8.3.1. Input Layer
 - 8.3.2. Hidden Layer
 - 8.3.3. Output Layer
- 8.4. Layer Bonding and Operations
 - 8.4.1. Architecture Design
 - 8.4.2. Connection between Layers
 - 8.4.3. Forward Propagation
- 8.5. Construction of the First Neural Network
 - 8.5.1. Network Design
 - 8.5.2. Establish the Weights
 - 8.5.3. Network Training
- 8.6. Trainer and Optimizer
 - 8.6.1. Optimizer Selection
 - 8.6.2. Establishment of a Loss Function
 - 8.6.3. Establishing a Metric
- 8.7. Application of the Principles of Neural Networks
 - 8.7.1. Activation Functions
 - 8.7.2. Backward Propagation
 - 8.7.3. Parameter Adjustment
- 8.8. From Biological to Artificial Neurons
 - 8.8.1. Functioning of a Biological Neuron
 - 8.8.2. Transfer of Knowledge to Artificial Neurons
 - 8.8.3. Establish Relations Between the Two

- 8.9. Implementation of MLP (Multilayer Perceptron) with Keras
 - 8.9.1. Definition of the Network Structure
 - 8.9.2. Model Compilation
 - 8.9.3. Model Training
- 8.10. Fine Tuning Hyperparameters of Neural Networks
 - 8.10.1. Selection of the Activation Function
 - 8.10.2. Set the Learning Rate
 - 8.10.3. Adjustment of Weights

Module 9. Deep Neural Networks Training

- 9.1. Gradient Problems
 - 9.1.1. Gradient Optimization Techniques
 - 9.1.2. Stochastic Gradients
 - 9.1.3. Weight Initialization Techniques
- 9.2. Reuse of Pre-Trained Layers
 - 9.2.1. Transfer Learning Training
 - 9.2.2. Feature Extraction
 - 9.2.3. Deep Learning
- 9.3. Optimizers
 - 9.3.1. Stochastic Gradient Descent Optimizers
 - 9.3.2. Adam and RMSprop Optimizers
 - 9.3.3. Moment Optimizers
- 9.4. Learning Rate Programming
 - 9.4.1. Automatic Learning Rate Control
 - 9.4.2. Learning Cycles
 - 9.4.3. Smoothing Terms
- 9.5. Overfitting
 - 9.5.1. Cross-Validation
 - 9.5.2. Regularization
 - 9.5.3. Evaluation Metrics
- 9.6. Practical Guidelines
 - 9.6.1. Model Design
 - 9.6.2. Selection of Metrics and Evaluation Parameters
 - 9.6.3. Hypothesis Testing

- 9.7. *Transfer Learning*
 - 9.7.1. Transfer Learning Training
 - 9.7.2. Feature Extraction
 - 9.7.3. Deep Learning

- 9.8. *Data Augmentation*
 - 9.8.1. Image Transformations
 - 9.8.2. Synthetic Data Generation
 - 9.8.3. Text Transformation

- 9.9. Practical Application of Transfer Learning
 - 9.9.1. Transfer Learning Training
 - 9.9.2. Feature Extraction
 - 9.9.3. Deep Learning

- 9.10. Regularization
 - 9.10.1. L and L
 - 9.10.2. Regularization by Maximum Entropy
 - 9.10.3. Dropout

Module 10. Model Customization and Training with TensorFlow

- 10.1. TensorFlow
 - 10.1.1. Using the TensorFlow Library
 - 10.1.2. Model Education with TensorFlow
 - 10.1.3. Operations with Graphs in TensorFlow
- 10.2. TensorFlow and NumPy
 - 10.2.1. NumPy Computational Environment for TensorFlow
 - 10.2.2. Using NumPy Arrays with TensorFlow
 - 10.2.3. NumPy Operations for TensorFlow Graphs
- 10.3. Model Customization and Training Algorithms
 - 10.3.1. Building Custom Models with TensorFlow
 - 10.3.2. Management of Training Parameters
 - 10.3.3. Use of Optimization Techniques for Training

- 10.4. TensorFlow Functions and Graphs
 - 10.4.1. Functions with TensorFlow
 - 10.4.2. Use of Graphs for Model Training
 - 10.4.3. Optimization of Graphs with TensorFlow Operations
- 10.5. Data Loading and Pre-Processing with TensorFlow
 - 10.5.1. Loading of Datasets with TensorFlow
 - 10.5.2. Data Pre-Processing with TensorFlow
 - 10.5.3. Using TensorFlow Tools for Data Manipulation
- 10.6. The tfdata API
 - 10.6.1. Using the tfdata API for Data Processing
 - 10.6.2. Construction of Data Streams with tf.data
 - 10.6.3. Using the tfdata API for Model Training
- 10.7. The TFRecord Format
 - 10.7.1. Using the TFRecord API for Data Serialization
 - 10.7.2. Loading TFRecord Files with TensorFlow
 - 10.7.3. Using TFRecord Files for Training Models
- 10.8. Keras Pre-Processing Layers
 - 10.8.1. Using the Keras Pre-Processing API
 - 10.8.2. Construction of Pre-Processing Pipelined with Keras
 - 10.8.3. Using the Keras Pre-Processing API for Model Training
- 10.9. The TensorFlow Datasets Project
 - 10.9.1. Using TensorFlow Datasets for Data Loading
 - 10.9.2. Data Pre-Processing with TensorFlow Datasets
 - 10.9.3. Using TensorFlow Datasets for Model Training
- 10.10. Building a Deep Learning App with TensorFlow
 - 10.10.1. Practical Application
 - 10.10.2. Building a Deep Learning App with TensorFlow
 - 10.10.3. Training a Model with TensorFlow
 - 10.10.4. Using the Application for the Prediction of Results

Module 11. Deep Computer Vision with Convolutional Neural Networks

- 11.1. The Visual Cortex Architecture
 - 11.1.1. Functions of the Visual Cortex
 - 11.1.2. Theories of Computational Vision
 - 11.1.3. Models of Image Processing
- 11.2. Convolutional Layers
 - 11.2.1. Reuse of Weights in Convolution
 - 11.2.2. Convolution D
 - 11.2.3. Activation Functions
- 11.3. Grouping Layers and Implementation of Grouping Layers with Keras
 - 11.3.1. Pooling and Striding
 - 11.3.2. *Flattening*
 - 11.3.3. Types of Pooling
- 11.4. CNN Architecture
 - 11.4.1. VGG Architecture
 - 11.4.2. AlexNet Architecture
 - 11.4.3. ResNet Architecture
- 11.5. Implementing a CNN ResNet- Using Keras
 - 11.5.1. Weight Initialization
 - 11.5.2. Input Layer Definition
 - 11.5.3. Output Definition
- 11.6. Use of Pre-Trained Keras Models
 - 11.6.1. Characteristics of Pre-Trained Models
 - 11.6.2. Uses of Pre-Trained Models
 - 11.6.3. Advantages of Pre-Trained Models
- 11.7. Pre-Trained Models for Transfer Learning
 - 11.7.1. Learning by Transfer
 - 11.7.2. Transfer Learning Process
 - 11.7.3. Advantages of Transfer Learning
- 11.8. Deep Computer Vision Classification and Localization
 - 11.8.1. Image Classification
 - 11.8.2. Localization of Objects in Images
 - 11.8.3. Object Detection

- 11.9. Object Detection and Object Tracking
 - 11.9.1. Object Detection Methods
 - 11.9.2. Object Tracking Algorithms
 - 11.9.3. Tracking and Localization Techniques
- 11.10. Semantic Segmentation
 - 11.10.1. Deep Learning for Semantic Segmentation
 - 11.10.2. Edge Detection
 - 11.10.3. Rule-Based Segmentation Methods

Module 12. Natural Language Processing (NLP) with Recurrent Neural Networks (RNN) and Attention

- 12.1. Text Generation Using RNN
 - 12.1.1. Training an RNN for Text Generation
 - 12.1.2. Natural Language Generation with RNN
 - 12.1.3. Text Generation Applications with RNN
- 12.2. Training Data Set Creation
 - 12.2.1. Preparation of the Data for Training an RNN
 - 12.2.2. Storage of the Training Dataset
 - 12.2.3. Data Cleaning and Transformation
 - 12.2.4. Sentiment Analysis
- 12.3. Classification of Opinions with RNN
 - 12.3.1. Detection of Themes in Comments
 - 12.3.2. Sentiment Analysis with Deep Learning Algorithms
- 12.4. Encoder-Decoder Network for Neural Machine Translation
 - 12.4.1. Training an RNN for Machine Translation
 - 12.4.2. Use of an Encoder-Decoder Network for Machine Translation
 - 12.4.3. Improving the Accuracy of Machine Translation with RNNs
- 12.5. Attention Mechanisms
 - 12.5.1. Application of Care Mechanisms in RNN
 - 12.5.2. Use of Care Mechanisms to Improve the Accuracy of the Models
 - 12.5.3. Advantages of Attention Mechanisms in Neural Networks

- 12.6. Transformer Models
 - 12.6.1. Using Transformers Models for Natural Language Processing
 - 12.6.2. Application of Transformers Models for Vision
 - 12.6.3. Advantages of Transformers Models
- 12.7. Transformers for Vision
 - 12.7.1. Use of Transformers Models for Vision
 - 12.7.2. Image Data Pre-Processing
 - 12.7.3. Training a Transformers Model for Vision
- 12.8. Hugging Face Transformer Library
 - 12.8.1. Using the Hugging Face Transformers Library
 - 12.8.2. Application of the Hugging Face Transformers Library
 - 12.8.3. Advantages of the Hugging Face Transformers Library
- 12.9. Other Transformers Libraries. Comparison
 - 12.9.1. Comparison Between Different Transformers Libraries
 - 12.9.2. Use of the Other Transformers Libraries
 - 12.9.3. Advantages of the Other Transformers Libraries
- 12.10. Development of an NLP Application with RNN and Attention. Practical Application
 - 12.10.1. Development of a Natural Language Processing Application with RNN and Attention
 - 12.10.2. Use of RNN, Attention Mechanisms and Transformers Models in the Application
 - 12.10.3. Evaluation of the Practical Application

Module 13. Autoencoders, GANs and Diffusion Models

- 13.1. Representation of Efficient Data
 - 13.1.1. Dimensionality Reduction
 - 13.1.2. Deep Learning
 - 13.1.3. Compact Representations
- 13.2. PCA Realization with an Incomplete Linear Automatic Encoder
 - 13.2.1. Training Process
 - 13.2.2. Implementation in Python
 - 13.2.3. Use of Test Data
- 13.3. Stacked Automatic Encoders
 - 13.3.1. Deep Neural Networks
 - 13.3.2. Construction of Coding Architectures
 - 13.3.3. Use of Regularization

- 13.4. Convolutional Autoencoders
 - 13.4.1. Design of Convolutional Models
 - 13.4.2. Convolutional Model Training
 - 13.4.3. Results Evaluation
- 13.5. Noise Suppression of Automatic Encoders
 - 13.5.1. Filter Application
 - 13.5.2. Design of Coding Models
 - 13.5.3. Use of Regularization Techniques
- 13.6. Sparse Automatic Encoders
 - 13.6.1. Increasing Coding Efficiency
 - 13.6.2. Minimizing the Number of Parameters
 - 13.6.3. Using Regularization Techniques
- 13.7. Variational Automatic Encoders
 - 13.7.1. Use of Variational Optimization
 - 13.7.2. Unsupervised Deep Learning
 - 13.7.3. Deep Latent Representations
- 13.8. Generation of Fashion MNIST Images
 - 13.8.1. Pattern Recognition
 - 13.8.2. Image Generation
 - 13.8.3. Deep Neural Networks Training
- 13.9. Generative Adversarial Networks and Diffusion Models
 - 13.9.1. Content Generation from Images
 - 13.9.2. Modeling of Data Distributions
 - 13.9.3. Use of Adversarial Networks
- 13.10. Implementation of the Models
 - 13.10.1. Practical Application
 - 13.10.2. Implementation of the Models
 - 13.10.3. Use of Real Data
 - 13.10.4. Results Evaluation

Module 14. Bio-Inspired Computing

- 14.1. Introduction to Bio-Inspired Computing
 - 14.1.1. Introduction to Bio-Inspired Computing
- 14.2. Social Adaptation Algorithms
 - 14.2.1. Bio-Inspired Computation Based on Ant Colonies
 - 14.2.2. Variants of Ant Colony Algorithms
 - 14.2.3. Particle Cloud Computing
- 14.3. Genetic Algorithms
 - 14.3.1. General Structure
 - 14.3.2. Implementations of the Major Operators
- 14.4. Space Exploration-Exploitation Strategies for Genetic Algorithms
 - 14.4.1. CHC Algorithm
 - 14.4.2. Multimodal Problems
- 14.5. Evolutionary Computing Models (I)
 - 14.5.1. Evolutionary Strategies
 - 14.5.2. Evolutionary Programming
 - 14.5.3. Algorithms Based on Differential Evolution
- 14.6. Evolutionary Computation Models (II)
 - 14.6.1. Evolutionary Models Based on Estimation of Distributions (EDA)
 - 14.6.2. Genetic Programming
- 14.7. Evolutionary Programming Applied to Learning Problems
 - 14.7.1. Rules-Based Learning
 - 14.7.2. Evolutionary Methods in Instance Selection Problems
- 14.8. Multi-Objective Problems
 - 14.8.1. Concept of Dominance
 - 14.8.2. Application of Evolutionary Algorithms to Multi-Objective Problems
- 14.9. Neural Networks (I)
 - 14.9.1. Introduction to Neural Networks
 - 14.9.2. Practical Example with Neural Networks
- 14.10. Neural Networks (II)
 - 14.10.1. Use Cases of Neural Networks in Medical Research
 - 14.10.2. Use Cases of Neural Networks in Economics
 - 14.10.3. Use Cases of Neural Networks in Artificial Vision

Module 15. Artificial Intelligence: Strategies and Applications

- 15.1. Financial Services
 - 15.1.1. The Implications of Artificial Intelligence (AI) in Financial Services. Opportunities and Challenges
 - 15.1.2. Use Cases
 - 15.1.3. Potential Risks Related to the Use of AI
 - 15.1.4. Potential Future Developments/Uses of AI
- 15.2. Implications of Artificial Intelligence in Healthcare Service
 - 15.2.1. Implications of AI in the Healthcare Sector. Opportunities and Challenges
 - 15.2.2. Use Cases
- 15.3. Risks Related to the Use of AI in Healthcare Service
 - 15.3.1. Potential Risks Related to the Use of AI
 - 15.3.2. Potential Future Developments/Uses of AI
- 15.4. *Retail*
 - 15.4.1. Implications of AI in Retail. Opportunities and Challenges
 - 15.4.2. Use Cases
 - 15.4.3. Potential Risks Related to the Use of AI
 - 15.4.4. Potential Future Developments/Uses of AI
- 15.5. Industry
 - 15.5.1. Implications of AI in Industry. Opportunities and Challenges
 - 15.5.2. Use Cases
- 15.6. Potential Risks Related to the Use of AI in Industry
 - 15.6.1. Use Cases
 - 15.6.2. Potential Risks Related to the Use of AI
 - 15.6.3. Potential Future Developments/Uses of AI
- 15.7. Public Administration
 - 15.7.1. AI Implications for Public Administration. Opportunities and Challenges
 - 15.7.2. Use Cases
 - 15.7.3. Potential Risks Related to the Use of AI
 - 15.7.4. Potential Future Developments/Uses of AI

- 15.8. Education
 - 15.8.1. AI Implications for Education. Opportunities and Challenges
 - 15.8.2. Use Cases
 - 15.8.3. Potential Risks Related to the Use of AI
 - 15.8.4. Potential Future Developments/Uses of AI
- 15.9. Forestry and Agriculture
 - 15.9.1. Implications of AI in Forestry and Agriculture. Opportunities and Challenges
 - 15.9.2. Use Cases
 - 15.9.3. Potential Risks Related to the Use of AI
 - 15.9.4. Potential Future Developments/Uses of AI
- 15.10. Human Resources
 - 15.10.1. Implications of AI in Human Resources. Opportunities and Challenges
 - 15.10.2. Use Cases
 - 15.10.3. Potential Risks Related to the Use of AI
 - 15.10.4. Potential Future Developments/Uses of AI

Module 16. Software Development Productivity Improvement with Artificial Intelligence

- 16.1. Preparing a Suitable Development Environment
 - 16.1.1. Essential Tool Selection for AI Development
 - 16.1.2. Configuration of the Selected Tools
 - 16.1.3. Implementation of CI/CD Pipelines Adapted to AI Projects
 - 16.1.4. Efficient Management of Dependencies and Versions in Development Environments
- 16.2. Essential AI Extensions for Visual Studio Code
 - 16.2.1. Exploring and Selecting AI Extensions for Visual Studio Code
 - 16.2.2. Integrating Static and Dynamic Analysis Tools into the Integrated Development Environment (IDE)
 - 16.2.3. Automation of Repetitive Tasks with Specific Extensions
 - 16.2.4. Customization of the Development Environment to Improve Efficiency
- 16.3. No-Code Design of User Interfaces with AI Elements
 - 16.3.1. No-Code Design Principles and their Application to User Interfaces
 - 16.3.2. Incorporation of AI Elements in Visual Interface Design
 - 16.3.3. Tools and Platforms for the No-Code Creation of Intelligent Interfaces
 - 16.3.4. Evaluation and Continuous Improvement of No-Code Interfaces with AI
- 16.4. Code Optimization Using ChatGPT
 - 16.4.1. Duplicate Code Detection
 - 16.4.2. Refactor
 - 16.4.3. Create Readable Code
 - 16.4.4. Understanding What Code Does
 - 16.4.5. Improving Variable and Function Naming
 - 16.4.6. Creating Automatic Documentation
- 16.5. Repository Management with AI
 - 16.5.1. Automation of Version Control Processes with AI Techniques
 - 16.5.2. Conflict Detection and Automatic Resolution in Collaborative Environments
 - 16.5.3. Predictive Analysis of Changes and Trends in Code Repositories
 - 16.5.4. Improvements in the Organization and Categorization of Repositories using AI
- 16.6. Integration of AI in Database Management
 - 16.6.1. Optimization of Queries and Performance Using AI Techniques
 - 16.6.2. Predictive Analysis of Database Access Patterns
 - 16.6.3. Implementation of Recommender Systems to Optimize Database Structure
 - 16.6.4. Proactive Monitoring and Detection of Potential Database Problems
- 16.7. Fault Detection and Creation of Unit Tests with AI ChatGPT
 - 16.7.1. Automatic Generation of Test Cases using AI Techniques
 - 16.7.2. Early Detection of Vulnerabilities and Bugs using Static Analysis with AI
 - 16.7.3. Improving Test Coverage by Identifying Critical Areas by AI
- 16.8. Pair Programming with GitHub Copilot
 - 16.8.1. Integration and Effective Use of GitHub Copilot in Pair Programming Sessions
 - 16.8.2. Integration Improvements in Communication and Collaboration among Developers with GitHub Copilot
 - 16.8.3. Integration Strategies to Maximize the Use of GitHub Copilot-Generated Code suggestions
 - 16.8.4. Integration Case Studies and Best Practices in AI-Assisted Pair Programming
- 16.9. Automatic Translation between Programming Languages ChatGPT
 - 16.9.1. Specific Machine Translation Tools and Services for Programming Languages
 - 16.9.2. Adaptation of Machine Translation Algorithms to Development Contexts
 - 16.9.3. Improvement of Interoperability between Different Languages by Machine Translation
 - 16.9.4. Assessment and Mitigation of Potential Challenges and Limitations in Machine Translation

- 16.10. Recommended AI Tools to Improve Productivity
 - 16.10.1. Comparative Analysis of AI Tools for Software Development
 - 16.10.2. Integration of AI Tools in Workflows
 - 16.10.3. Automation of Routine Tasks with AI Tools
 - 16.10.4. Evaluation and Selection of Tools Based on Project Context and Requirements

Module 17. Software Architecture with Artificial Intelligence

- 17.1. Optimization and Performance Management in AI Tools with the Help of ChatGPT
 - 17.1.1. Performance Analysis and Profiling in AI Tools
 - 17.1.2. Algorithm Optimization Strategies and AI Models
 - 17.1.3. Implementation of Caching and Parallelization Techniques to Improve Performance
 - 17.1.4. Tools and Methodologies for Continuous Real-Time Performance Monitoring
- 17.2. Scalability in AI Applications Using ChatGPT
 - 17.2.1. Scalable Architectures Design for AI Applications
 - 17.2.2. Implementation of Partitioning and Load Sharing Techniques
 - 17.2.3. Workflow and Workload Management in Scalable Systems
 - 17.2.4. Strategies for Horizontal and Vertical Expansion in Variable Demand Environments
- 17.3. Maintainability of AI Applications Using ChatGPT
 - 17.3.1. Design Principles to Facilitate Maintainability in AI Projects
 - 17.3.2. Specific Documentation Strategies for AI Models and Algorithms
 - 17.3.3. Implementation of Unit and Integration Tests to Facilitate Maintainability
 - 17.3.4. Methods for Refactoring and Continuous Improvement in Systems with AI Components
- 17.4. Large-Scale System Design
 - 17.4.1. Architectural Principles for Large-Scale System Design
 - 17.4.2. Decomposition of Complex Systems into Microservices
 - 17.4.3. Implementation of Specific Design Patterns for Distributed Systems
 - 17.4.4. Strategies for Complexity Management in Large-Scale Architectures with AI Components
- 17.5. Large-Scale Data Warehousing for AI Tools
 - 17.5.1. Selection of Scalable Data Storage Technologies
 - 17.5.2. Design of Database Schemas for Efficient Handling of Large Data Volumes
 - 17.5.3. Partitioning and Replication Strategies in Massive Data Storage Environments
 - 17.5.4. Implementation of Data Management Systems to Ensure Integrity and Availability in AI Projects
- 17.6. Data Structures with AI Using ChatGPT
 - 17.6.1. Adaptation of Classical Data Structures for Use with AI Algorithms
 - 17.6.2. Design and Optimization of Specific Data Structures with ChatGPT
 - 17.6.3. Integration of Efficient Data Structures in Data Intensive Systems
 - 17.6.4. Strategies for Real-Time Data Manipulation and Storage in AI Data Structures
- 17.7. Programming Algorithms for AI Products
 - 17.7.1. Development and Implementation of Application-Specific Algorithms for AI Applications
 - 17.7.2. Algorithm Selection Strategies according to Problem Type and Product Requirements
 - 17.7.3. Adaptation of Classical Algorithms for Integration into AI Systems
 - 17.7.4. Evaluation and Performance Comparison between Different Algorithms in Development Contexts with AI
- 17.8. Design Patterns for AI Development
 - 17.8.1. Identification and Application of Common Design Patterns in Projects with AI Components
 - 17.8.2. Development of Specific Patterns for the Integration of Models and Algorithms into Existing Systems
 - 17.8.3. Strategies for the Implementation of Patterns to Improve Reusability and Maintainability in AI Projects
 - 17.8.4. Case Studies and Best Practices in the Application of Design Patterns in AI Architectures
- 17.9. Implementation of Clean Architecture using ChatGPT
 - 17.9.1. Fundamental Principles and Concepts of Clean Architecture
 - 17.9.2. Adaptation of Clean Architecture to Projects with AI Components
 - 17.9.3. Implementation of Layers and Dependencies in Systems with Clean Architecture
 - 17.9.4. Benefits and Challenges of Implementing Clean Architecture in Software Development with AI
- 17.10. Secure Software Development in Web Applications with DeepCode
 - 17.10.1. Principles of Security in the Development of Software with AI Components
 - 17.10.2. Identification and Mitigation of Potential Vulnerabilities in AI Models and Algorithms
 - 17.10.3. Implementation of Secure Development Practices in Web Applications with Artificial Intelligence Functionalities
 - 17.10.4. Strategies for the Protection of Sensitive Data and Prevention of Attacks in AI Projects

Module 18. Web Projects with Artificial Intelligence

- 18.1. Working Environment Preparation for Web Development with AI
 - 18.1.1. Configuration of Web Development Environments for Projects with Artificial Intelligence
 - 18.1.2. Selection and Preparation of Essential Tools for Web Development with AI
 - 18.1.3. Integration of Specific Libraries and Frameworks for Web Projects with Artificial Intelligence
 - 18.1.4. Implementation of Best Practices in the Configuration of Collaborative Development Environments
- 18.2. Workspace Creation for AI Projects
 - 18.2.1. Effective Design and Organization of Workspaces for Web Projects with Artificial Intelligence Components
 - 18.2.2. Use of Project Management and Version Control Tools in the Workspace
 - 18.2.3. Strategies for Efficient Collaboration and Communication in the Development Team
 - 18.2.4. Adaptation of the Workspace to the Specific Needs of AI Web Projects
- 18.3. Design Patterns in Github Copilot Products
 - 18.3.1. Identification and Application of Common Design Patterns in User Interfaces with Artificial Intelligence Elements
 - 18.3.2. Development of Specific Patterns to Improve the User Experience in AI Web Projects
 - 18.3.3. Integration of Design Patterns in the Overall Architecture of Web Projects with Artificial Intelligence
 - 18.3.4. Evaluation and Selection of Appropriate Design Patterns According to the Project's Context
- 18.4. Front-End Development with GitHub Copilot
 - 18.4.1. Integration of AI Models in the Presentation Layer of Web Projects
 - 18.4.2. Development of Adaptive User Interfaces with Artificial Intelligence Elements
 - 18.4.3. Implementation of Natural Language Processing (NLP) Functionalities in Frontend Development
 - 18.4.4. Strategies for Performance Optimization in Front-End Development with AI
- 18.5. Database Creation using GitHub Copilot
 - 18.5.1. Selection of Database Technologies for Web Projects with Artificial Intelligence
 - 18.5.2. Design of Database Schemas for Storing and Managing AI-Related Data
 - 18.5.3. Implementation of Efficient Storage Systems for Large Volumes of Data Generated by AI Models
 - 18.5.4. Strategies for Security and Protection of Sensitive Data in AI Web Project Databases
- 18.6. Back-End Development with GitHub Copilot
 - 18.6.1. Integration of AI Services and Models in the Back-End Business Logic
 - 18.6.2. Development of Specific APIs and Endpoints for Communication between Front-End and AI Components
 - 18.6.3. Implementation of Data Processing and Decision-Making Logic in the Backend with Artificial Intelligence
 - 18.6.4. Strategies for Scalability and Performance in Back-End Development of Web Projects with AI
- 18.7. Optimization of the Deployment Process of Your Website
 - 18.7.1. Automation of Web Project Build and Deployment Processes with ChatGPT
 - 18.7.2. Implementing CI/CD Pipelines Tailored to Web Applications with Github Copilot
 - 18.7.3. Strategies for Efficient Release and Upgrade Management in Continuous Deployments
 - 18.7.4. Post-Deployment Monitoring and Analysis for Continuous Process Improvement
- 18.8. AI in Cloud Computing
 - 18.8.1. Integration of Artificial Intelligence Services in Cloud Computing Platforms
 - 18.8.2. Development of Scalable and Distributed Solutions using Cloud Services with AI Capabilities
 - 18.8.3. Strategies for Efficient Resource and Cost Management in Cloud Environments with AI-enabled Web Applications
 - 18.8.4. Evaluation and Comparison of Cloud Service Providers for AI-enabled Web Projects
- 18.9. Creating an AI Project for LAMP Environments with the Help of ChatGPT
 - 18.9.1. Adaptation of Web Projects Based on the LAMP Stack to Include Artificial Intelligence Components
 - 18.9.2. Integration of AI-specific Libraries and Frameworks in LAMP Environments
 - 18.9.3. Development of AI Functionalities that Complement the Traditional LAMP Architecture
 - 18.9.4. Strategies for Optimization and Maintenance in Web Projects with AI in LAMP Environments
- 18.10. Creating an AI Project for MEVN Environments Using ChatGPT
 - 18.10.1. Integration of MEVN Stack Technologies and Tools with Artificial Intelligence Components
 - 18.10.2. Development of Modern and Scalable Web Applications in MEVN Environments with AI Capabilities
 - 18.10.3. Implementation of Data Processing and Machine Learning Functionalities in MEVN Projects
 - 18.10.4. Strategies for Performance and Security Enhancement of AI-enabled Web Applications in MEVN Environments

Module 19. Mobile Applications with Artificial Intelligence

- 19.1. Working Environment Preparation for Mobile Development with AI
 - 19.1.1. Configuration of Mobile Development Environments for Projects with Artificial Intelligence
 - 19.1.2. Selection and Preparation of Specific Tools for Mobile Application Development with AI
 - 19.1.3. Integration of AI-Libraries and Frameworks in Mobile Development Environments
 - 19.1.4. Configuration of Emulators and Real Devices for Testing Mobile Applications with AI Components
- 19.2. Creation of a Workspace with GitHub Copilot
 - 19.2.1. Integration of GitHub Copilot in Mobile Development Environments
 - 19.2.2. Effective Use of GitHub Copilot for Code Generation in AI Projects
 - 19.2.3. Strategies for Developer Collaboration when Using GitHub Copilot in the Workspace
 - 19.2.4. Best Practices and Limitations in the Use of GitHub Copilot in Mobile Application Development with AI
- 19.3. Firebase Configuration
 - 19.3.1. Initial Configuration of a Firebase Project for Mobile Development
 - 19.3.2. Firebase Integration in Mobile Applications with Artificial Intelligence Functionality
 - 19.3.3. Use of Firebase Services as Database, Authentication, and Notifications in AI projects
 - 19.3.4. Strategies for Real-Time Data and Event Management in Firebase-enabled Mobile Applications
- 19.4. Concepts of Clean Architecture, DataSources, Repositories
 - 19.4.1. Fundamental Principles of Clean Architecture in Mobile Development with AI
 - 19.4.2. Implementation of DataSources and Repositories Layers with GitHub Copilot
 - 19.4.3. Design and Structuring of Components in Mobile Projects with Github Copilot
 - 19.4.4. Benefits and Challenges of Implementing Clean Architecture in Mobile Applications with AI
- 19.5. Creating Authentication Screen with GitHub Copilot
 - 19.5.1. Design and Development of User Interfaces for Authentication Screens in Mobile Applications with IA
 - 19.5.2. Integration of Authentication Services with Firebase in the Login Screen
 - 19.5.3. Use of Security and Data Protection Techniques in the Authentication Screen
 - 19.5.4. Personalization and Customization of the User Experience in the Authentication Screen
- 19.6. Creating Dashboard and Navigation with GitHub Copilot
 - 19.6.1. Dashboard Design and Development with Artificial Intelligence Elements
 - 19.6.2. Implementation of Efficient Navigation Systems in Mobile Applications with AI
 - 19.6.3. Integration of AI Functionalities in the Dashboard to Improve User Experience
- 19.7. Listing Screen Creation using GitHub Copilot
 - 19.7.1. Development of User Interfaces for Listing Screens in AI-enabled Mobile Applications
 - 19.7.2. Integration of Recommendation and Filtering Algorithms into the Listing Screen
 - 19.7.3. Use of Design Patterns for Effective Presentation of Data in the Listing Screen
 - 19.7.4. Strategies for Efficient Loading of Real-Time Data into the Listing Screen
- 19.8. Creating Details Screen with GitHub Copilot
 - 19.8.1. Design and Development of Detailed User Interfaces for the Presentation of Specific Information
 - 19.8.2. Integration of AI Functionalities to Enrich the Detailed Screen
 - 19.8.3. Implementation of Interactions and Animations in the Detailed Screen
 - 19.8.4. Strategies for Performance Optimization in Loading and Detail Display in AI-enabled Mobile Applications
- 19.9. Creating a Settings Screen with GitHub Copilot
 - 19.9.1. Development of User Interfaces for Configuration and Settings in AI-enabled Mobile Applications
 - 19.9.2. Integration of Customized Settings Related to Artificial Intelligence Components
 - 19.9.3. Implementation of Customized Options and Preferences in the Settings Screen
 - 19.9.4. Strategies for Usability and Clarity in the Presentation of Options in the Settings Screen
- 19.10. Creation of Icons, Splash and Graphic Resources for Your App with AI
 - 19.10.1. Design and Creation of Attractive Icons to Represent the AI Mobile Application
 - 19.10.2. Development of Splash Screens with Impactful Visuals
 - 19.10.3. Selection and Adaptation of Graphic Resources to Enhance the Aesthetics of the Mobile Application
 - 19.10.4. Strategies for Consistency and Visual Branding in the Graphic Elements of the Application with AI

Module 20. Artificial Intelligence for QA Testing

- 20.1. Software Testing Life Cycle
 - 20.1.1. Description and Understanding of the Testing Life Cycle in Software Development
 - 20.1.2. Phases of the Testing Life Cycle and its Importance in Quality Assurance
 - 20.1.3. Integration of Artificial Intelligence in Different Stages of the Testing Life Cycle
 - 20.1.4. Strategies for Continuous Improvement of the Testing Life Cycle using AI
- 20.2. Test Cases and Bug Detection with the Help of ChatGPT
 - 20.2.1. Effective Test Case Design and Writing in the Context of QA Testing
 - 20.2.2. Identification of Bugs and Errors during Test Case Execution
 - 20.2.3. Application of Early Bug Detection Techniques using Static Analysis
 - 20.2.4. Use of Artificial Intelligence Tools for the Automatic Identification of Bugs in Test Cases
- 20.3. Types of Testing
 - 20.3.1. Exploration of Different Types of Testing in the QA Environment
 - 20.3.2. Unit, Integration, Functional, and Acceptance Testing: Characteristics and Applications
 - 20.3.3. Strategies for the Selection and Appropriate Combination of Testing Types in Projects with ChatGPT
 - 20.3.4. Adaptation of Conventional Testing Types to Projects with ChatGPT
- 20.4. Creation of a Testing Plan Using ChatGPT
 - 20.4.1. Design and Structure of a Comprehensive Testing Plan
 - 20.4.2. Identification of Requirements and Test Scenarios in AI Projects
 - 20.4.3. Strategies for Manual and Automated Test Planning
 - 20.4.4. Continuous Evaluation and Adjustment of the Testing Plan as the Project Develops
- 20.5. AI Bug Detection and Reporting
 - 20.5.1. Implementation of Automatic Bug Detection Techniques Using Machine Learning Algorithms
 - 20.5.2. Use of ChatGPT for Dynamic Code Analysis to Search for Possible Bugs
 - 20.5.3. Strategies for Automatic Generation of Detailed Reports on Bugs Detected Using ChatGPT
 - 20.5.4. Effective Collaboration between Development and QA Teams in the Management of AI-Detected Bugs
- 20.6. Creation of Automated Testing with AI
 - 20.6.1. Development of Automated Test Scripts for Projects Using ChatGPT
 - 20.6.2. Integration of AI-Based Test Automation Tools
 - 20.6.3. Using ChatGPT for Dynamic Generation of Automated Test Cases
 - 20.6.4. Strategies for Efficient Execution and Maintenance of Automated Test Cases in AI Projects
- 20.7. API Testing
 - 20.7.1. Fundamental Concepts of API Testing and its Importance in QA
 - 20.7.2. Development of Tests for the Verification of APIs in Environments Using ChatGPT
 - 20.7.3. Strategies for Data and Results Validation in API Testing with ChatGPT
 - 20.7.4. Use of Specific Tools for API Testing in Projects with Artificial Intelligence
- 20.8. AI Tools for Web Testing
 - 20.8.1. Exploration of Artificial Intelligence Tools for Test Automation in Web Environments
 - 20.8.2. Integration of Element Recognition and Visual Analysis Technologies in Web Testing
 - 20.8.3. Strategies for Automatic Detection of Changes and Performance Problems in Web Applications Using ChatGPT
 - 20.8.4. Evaluation of Specific Tools for Improving Efficiency in Web Testing with AI
- 20.9. Mobile Testing Using AI
 - 20.9.1. Development of Testing Strategies for Mobile Applications with AI Components
 - 20.9.2. Integration of Specific Testing Tools for AI-Based Mobile Platforms
 - 20.9.3. Use of ChatGPT for Detecting Performance Problems in Mobile Applications
 - 20.9.4. Strategies for the Validation of Interfaces and Specific Functions of Mobile Applications by AI
- 20.10. QA Tools with AI
 - 20.10.1. Exploration of QA Tools and Platforms that Incorporate Artificial Intelligence Functionality
 - 20.10.2. Evaluation of Tools for Efficient Test Management and Test Execution in AI Projects
 - 20.10.3. Using ChatGPT for the Generation and Optimization of Test Cases
 - 20.10.4. Strategies for Effective Selection and Adoption of QA Tools with AI Capabilities

04

Teaching Objectives

This university degree will provide specialists with the keys to implementing Artificial Intelligence solutions in the field of Programming. Graduates will acquire a comprehensive vision, combining the most up-to-date knowledge with practical skills that will improve their decision making. In turn, professionals will master the most modern tools for the development of software powered by machine learning. In this way, students will design proposals for both websites and mobile applications with adaptability. This will make them highly specialized to meet the current demands of a variety of industries.



“

You will master the security fundamentals in the use of Artificial Intelligence, guaranteeing the responsible development of algorithms in the institutional context”



General Objectives

- ♦ Develop skills to configure and manage efficient development environments, ensuring a solid foundation for the implementation of Artificial Intelligence projects
- ♦ Acquire skills in planning, executing and automating quality testing, incorporating Artificial Intelligence tools for bug detection and correction
- ♦ Understand and apply performance, scalability and maintainability principles in the design of large-scale computing systems
- ♦ Become familiar with the most important design patterns and apply them effectively in software architecture



You will delve into the integration of AI elements in Visual Studio Code and code optimization with ChatGPT”





Specific Objectives

Module 1. Fundamentals of Artificial Intelligence

- Analyze the historical evolution of Artificial Intelligence, from its beginnings to its current state, identifying key milestones and developments
- Understand the functioning of neural networks and their application in learning models in Artificial Intelligence
- Study the principles and applications of genetic algorithms, analyzing their usefulness in solving complex problems
- Analyze the importance of thesauri, vocabularies and taxonomies in the structuring and processing of data for AI systems
- Explore the concept of the semantic web and its influence on the organization and understanding of information in digital environments

Module 2. Data Types and Life Cycle

- Understand the fundamental concepts of statistics and their application in data analysis
- Identify and classify the different types of statistical data, from quantitative to qualitative data
- Analyze the life cycle of data, from generation to disposal, identifying key stages
- Explore the initial stages of the data life cycle, highlighting the importance of data planning and structure
- Study data collection processes, including methodology, tools and collection channels
- Explore the Datawarehouse concept, with emphasis on the elements that comprise it and its design
- Analyze the regulatory aspects related to data management, complying with privacy and security regulations, as well as best practices

Module 3. Data in Artificial Intelligence

- Master the fundamentals of data science, covering tools, types and sources for information analysis
- Explore the process of transforming data into information using data mining and visualization techniques
- Study the structure and characteristics of datasets, understanding their importance in the preparation and use of data for Artificial Intelligence models
- Analyze supervised and unsupervised models, including methods and classification
- Use specific tools and best practices in data handling and processing, ensuring efficiency and quality in the implementation of Artificial Intelligence

Module 4. Data Mining. Selection, Pre-Processing and Transformation

- Master the techniques of statistical inference to understand and apply statistical methods in data mining
- Perform detailed exploratory analysis of data sets to identify relevant patterns, anomalies, and trends
- Develop skills for data preparation, including data cleaning, integration, and formatting for use in data mining
- Implement effective strategies for handling missing values in datasets, applying imputation or elimination methods according to context
- Identify and mitigate noise present in data, using filtering and smoothing techniques to improve the quality of the data set
- Address data preprocessing in Big Data environments

Module 5. Algorithm and Complexity in Artificial Intelligence

- ♦ Introduce algorithm design strategies, providing a solid understanding of fundamental approaches to problem solving
- ♦ Analyze the efficiency and complexity of algorithms, applying analysis techniques to evaluate performance in terms of time and space
- ♦ Study and apply sorting algorithms, understanding their performance and comparing their efficiency in different contexts
- ♦ Explore tree-based algorithms, understanding their structure and applications
- ♦ Investigate algorithms with Heaps, analyzing their implementation and usefulness in efficient data manipulation
- ♦ Analyze graph-based algorithms, exploring their application in the representation and solution of problems involving complex relationships
- ♦ Study Greedy algorithms, understanding their logic and applications in solving optimization problems
- ♦ Investigate and apply the backtracking technique for systematic problem solving, analyzing its effectiveness in various scenarios

Module 6. Intelligent Systems

- ♦ Explore agent theory, understanding the fundamental concepts of its operation and its application in Artificial Intelligence and software engineering
- ♦ Study the representation of knowledge, including the analysis of ontologies and their application in the organization of structured information
- ♦ Analyze the concept of the semantic web and its impact on the organization and retrieval of information in digital environments

- ♦ Evaluate and compare different knowledge representations, integrating these to improve the efficiency and accuracy of intelligent systems
- ♦ Study semantic reasoners, knowledge-based systems and expert systems, understanding their functionality and applications in intelligent decision making

Module 7. Machine Learning and Data Mining

- ♦ Introduce the processes of knowledge discovery and the fundamental concepts of machine learning
- ♦ Study decision trees as supervised learning models, understanding their structure and applications
- ♦ Evaluate classifiers using specific techniques to measure their performance and accuracy in data classification
- ♦ Study neural networks, understanding their operation and architecture to solve complex machine learning problems
- ♦ Explore Bayesian methods and their application in machine learning, including Bayesian networks and Bayesian classifiers
- ♦ Analyze regression and continuous response models for predicting numerical values from data
- ♦ Study clustering techniques to identify patterns and structures in unlabeled data sets
- ♦ Explore text mining and natural language processing (NLP), understanding how machine learning techniques are applied to analyze and understand text

Module 8. Neural Networks, the Basis of Deep Learning

- ♦ Master the fundamentals of Deep Learning, understanding its essential role in Deep Learning
- ♦ Explore the fundamental operations in neural networks and understand their application in model building
- ♦ Analyze the different layers used in neural networks and learn how to select them appropriately
- ♦ Understand the effective linking of layers and operations to design complex and efficient neural network architectures
- ♦ Use trainers and optimizers to tune and improve the performance of neural networks
- ♦ Explore the connection between biological and artificial neurons for a deeper understanding of model design
- ♦ Tune hyperparameters for Fine Tuning of neural networks, optimizing their performance on specific tasks

Module 9. Deep Neural Networks Training

- ♦ Solve gradient-related problems in deep neural network training
- ♦ Explore and apply different optimizers to improve the efficiency and convergence of models
- ♦ Program the learning rate to dynamically adjust the convergence speed of the model
- ♦ Understand and address overfitting through specific strategies during training
- ♦ Apply practical guidelines to ensure efficient and effective training of deep neural networks
- ♦ Implement Transfer Learning as an advanced technique to improve model performance on specific tasks
- ♦ Explore and apply Data Augmentation techniques to enrich datasets and improve model generalization
- ♦ Understand and apply regularization techniques to improve generalization and avoid overfitting in deep neural networks

Module 10. Model Customization and Training with TensorFlow

- ♦ Master the fundamentals of TensorFlow and its integration with NumPy for efficient data management and calculations
- ♦ Customize models and training algorithms using the advanced capabilities of TensorFlow
- ♦ Explore the tfdata API to efficiently manage and manipulate datasets
- ♦ Implement the TFRecord format for storing and accessing large datasets in TensorFlow
- ♦ Use Keras preprocessing layers to facilitate the construction of custom models
- ♦ Explore the TensorFlow Datasets project to access predefined datasets and improve development efficiency
- ♦ Develop a Deep Learning application with TensorFlow, integrating the knowledge acquired in the module
- ♦ Apply in a practical way all the concepts learned in building and training custom models with TensorFlow in real-world situations

Module 11. Deep Computer Vision with Convolutional Neural Networks

- ♦ Understand the architecture of the visual cortex and its relevance in Deep Computer Vision
- ♦ Explore and apply convolutional layers to extract key features from images
- ♦ Implement clustering layers and their use in Deep Computer Vision models with Keras
- ♦ Develop and implement a CNN ResNet using the Keras library to improve model efficiency and performance
- ♦ Use pre-trained Keras models to leverage transfer learning for specific tasks
- ♦ Apply classification and localization techniques in Deep Computer Vision environments
- ♦ Explore object detection and object tracking strategies using Convolutional Neural Networks
- ♦ Implement semantic segmentation techniques to understand and classify objects in images in a detailed manner

Module 12. Natural Language Processing (NLP) with Recurrent Neural Networks (RNN) and Attention

- ♦ Develop skills in text generation using Recurrent Neural Networks
- ♦ Apply RNNs in opinion classification for sentiment analysis in texts
- ♦ Understand and apply attentional mechanisms in natural language processing models
- ♦ Analyze and use Transformers models in specific NLP tasks
- ♦ Explore the application of Transformers models in the context of image processing and computer vision
- ♦ Compare different Transformers libraries to evaluate their suitability for specific tasks
- ♦ Develop a practical application of NLP that integrates RNN and attention mechanisms to solve real-world problems

Module 13. Autoencoders, GANs and Diffusion Models

- ♦ Develop efficient representations of data using Autoencoders, GANs and Diffusion Models
- ♦ Perform PCA using an incomplete linear autoencoder to optimize data representation
- ♦ Implement and understand the operation of stacked autoencoders
- ♦ Explore and apply convolutional autoencoders for efficient visual data representations
- ♦ Analyze and apply the effectiveness of sparse automatic encoders in data representation
- ♦ Generate fashion images from the MNIST dataset using Autoencoders
- ♦ Understand the concept of Generative Adversarial Networks and Diffusion Models

Module 14. Bio-Inspired Computing

- ♦ Introduce the fundamental concepts of bio-inspired computing
- ♦ Explore social adaptation algorithms as a key approach in bio-inspired computing
- ♦ Analyze space exploration-exploitation strategies in genetic algorithms
- ♦ Examine models of evolutionary computation in the context of optimization
- ♦ Continue detailed analysis of evolutionary computation models
- ♦ Apply evolutionary programming to specific learning problems
- ♦ Address the complexity of multi-objective problems in the framework of bio-inspired computing
- ♦ Explore the application of neural networks in the field of bio-inspired computing
- ♦ Delve into the implementation and usefulness of neural networks in bio-inspired computing

Module 15. Artificial Intelligence: Strategies and Applications

- ♦ Develop strategies for the implementation of artificial intelligence in financial services
- ♦ Analyze the implications of artificial intelligence in the delivery of healthcare services
- ♦ Identify and assess the risks associated with the use of AI in the healthcare field
- ♦ Assess the potential risks associated with the use of AI in industry
- ♦ Apply artificial intelligence techniques in industry to improve productivity
- ♦ Design artificial intelligence solutions to optimize processes in public administration
- ♦ Evaluate the implementation of AI technologies in the education sector
- ♦ Apply artificial intelligence techniques in forestry and agriculture to improve productivity
- ♦ Optimize human resources processes through the strategic use of artificial intelligence

Module 16. Software Development Productivity Improvement with Artificial Intelligence

- ♦ Delve into the implementation of must-have AI extensions in Visual Studio Code to improve productivity and facilitate software development
- ♦ Gain a solid understanding of basic AI concepts and their application in software development, including machine learning algorithms, natural language processing, neural networks, etc
- ♦ Master the configuration of optimized development environments, ensuring that students are able to create environments conducive to AI projects
- ♦ Apply specific techniques using ChatGPT for the automatic identification and correction of potential code improvements, encouraging more efficient programming practices
- ♦ Promote collaboration between professionals from different programmers (from programmers to data engineers to user experience designers) to develop effective and ethical AI software solutions

Module 17. Software Architecture for QA Testing

- ♦ Develop skills to design solid test plans, covering different types of testing and ensuring software quality
- ♦ Recognize and analyze different types of software frameworks, such as monolithic, microservices or service oriented
- ♦ Gain a comprehensive vision on the principles and techniques for designing computer systems that are scalable and capable of handling large volumes of data
- ♦ Apply advanced skills in the implementation of AI-powered data structures to optimize software performance and efficiency
- ♦ Develop secure development practices, with a focus on avoiding vulnerabilities to ensure software security at the architectural level

Module 18. Web Projects with Artificial Intelligence

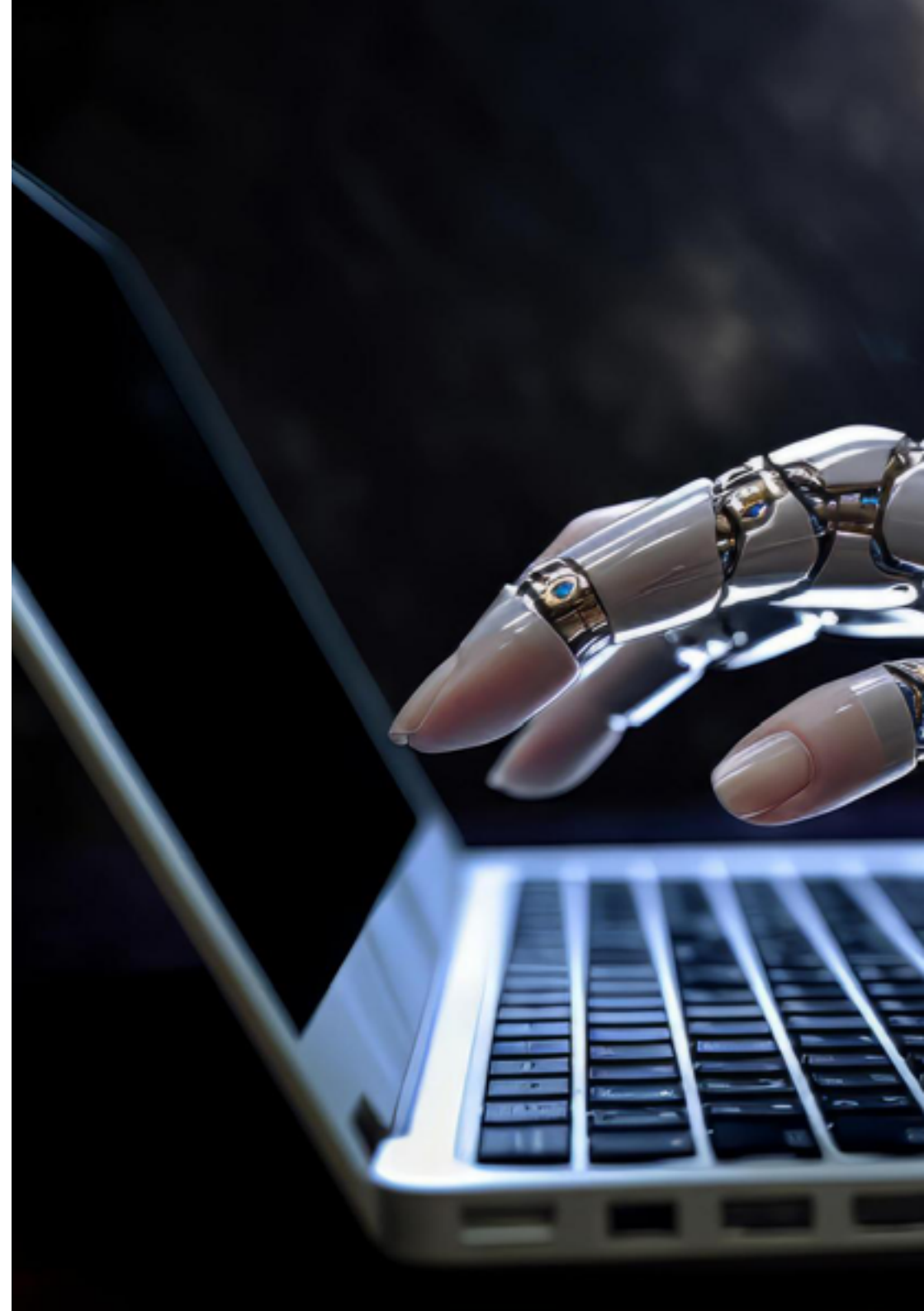
- ♦ Develop comprehensive skills for the implementation of web projects, from frontend design to backend optimization, with the inclusion of AI elements
- ♦ Optimize the process of deploying websites, incorporating techniques and tools to improve speed and efficiency
- ♦ Integrate AI into cloud computing, enabling students to create highly scalable and efficient web projects
- ♦ Acquire the ability to identify specific problems and opportunities in web projects where AI can be effectively applied, such as in text processing, personalization, content recommendation, etc
- ♦ Encourage students to keep abreast of the latest trends and advances in AI for its proper application in web projects

Module 19. Mobile Applications with Artificial Intelligence

- ♦ Apply advanced concepts of clean architecture, datasources and repositories to ensure a robust and modular structure in AI-enabled mobile applications
- ♦ Develop skills to design interactive screens, icons and graphical resources using AI to enhance the user experience in mobile applications
- ♦ Delve into the configuration of the mobile app framework and use Github Copilot to streamline the development process
- ♦ Optimize mobile applications with AI for efficient performance, taking into account resource management and data usage
- ♦ Perform quality testing of AI mobile applications, enabling students to identify problems and debug bugs

Module 20. Artificial Intelligence for QA Testing

- ♦ Master principles and techniques for designing computer systems that are scalable and capable of handling large volumes of data
- ♦ Apply advanced skills in the implementation of AI-powered data structures to optimize software performance and efficiency
- ♦ Understand and apply secure development practices, with a focus on avoiding vulnerabilities such as injection, to ensure software security at the architectural level
- ♦ Generate automated tests, especially in web and mobile environments, integrating AI tools to improve process efficiency
- ♦ Use advanced AI-powered QA tools for more efficient bug detection and continuous software improvement





“

You will delve into the integration of AI elements in Visual Studio Code and code optimization with ChatGPT”

05

Career Opportunities

This TECH university program represents a unique opportunity for IT professionals looking to update their skills and master Artificial Intelligence tools applied to the development of software, mobile applications and big data. Through innovative learning, graduates will be able to design, optimize and implement advanced solutions, improving the efficiency and scalability of intelligent systems. They will also be trained to lead technological projects, automate processes and apply AI in decision-making. With this cutting-edge knowledge, they will significantly expand their job opportunities in a highly competitive and constantly evolving sector.



```
// tooltipManager
private TooltipManager tooltipManager;
private Builder mTooltipContact;
private Builder mTooltipPassport;
private Builder mTooltipEmailTooltip;
private Builder mTooltipBillingTooltip;
private static final int TOOL TIP_ID_CONTACT = 101;
private static final int TOOL TIP_ID_PASSPORT = 102;
private static final int TOOL TIP_ID_EMAIL = 103;
private static final int TOOL TIP_ID_BILLING = 104;

private void mTooltipManager() {
    mTooltipContact = tooltipManager.create(this, TOOL TIP_ID_CONTACT)
        .anchor(findViewById(R.id.btnContact), TooltipManager.Gravity.BOTTOM)
        .closePolicy(TooltipManager.ClosePolicy.TouchOutside, 3000)
        .activateDelay(100)
        .fitToScreen(true)
        .withStyle(R.style.TooltipLayoutStyle)
        .text(R.string.label_tips_contact);
    mTooltipContact.show();
}

@Override
public void onBackPressed() {
    super.onBackPressed();
}

// validationRegisterFore() {
```

“

You will apply Artificial Intelligence technologies in software development and lead innovative projects in web and mobile environments"

Graduate Profile

Graduates of this TECH Master's Degree will be highly skilled professionals capable of integrating Artificial Intelligence technologies into software development, optimizing processes and improving the scalability of intelligent systems. They will also have advanced skills in designing, implementing and evaluating machine learning models, as well as developing web and mobile applications with integrated AI. In addition, they will master strategies to improve productivity in development environments, ensuring code quality through advanced QA testing and workflow optimization.

You will be able to stand out as a specialist in Artificial Intelligence applied to software development, leading innovation projects in digital and business environments.

- ♦ **Technological Innovation in Software Development:** Ability to integrate Artificial Intelligence tools into software projects, improving the efficiency and scalability of systems
- ♦ **Optimization of Intelligent Processes:** Ability to apply AI in the automation and improvement of workflows, optimizing performance in web, mobile and big data environments
- ♦ **Security and Ethics in AI:** Responsibility in the application of regulations and good practices to guarantee privacy and data protection in projects with Artificial Intelligence
- ♦ **Collaboration in Technological Environments:** Aptitude to work in multidisciplinary teams, facilitating the integration of AI in the development of innovative digital solutions





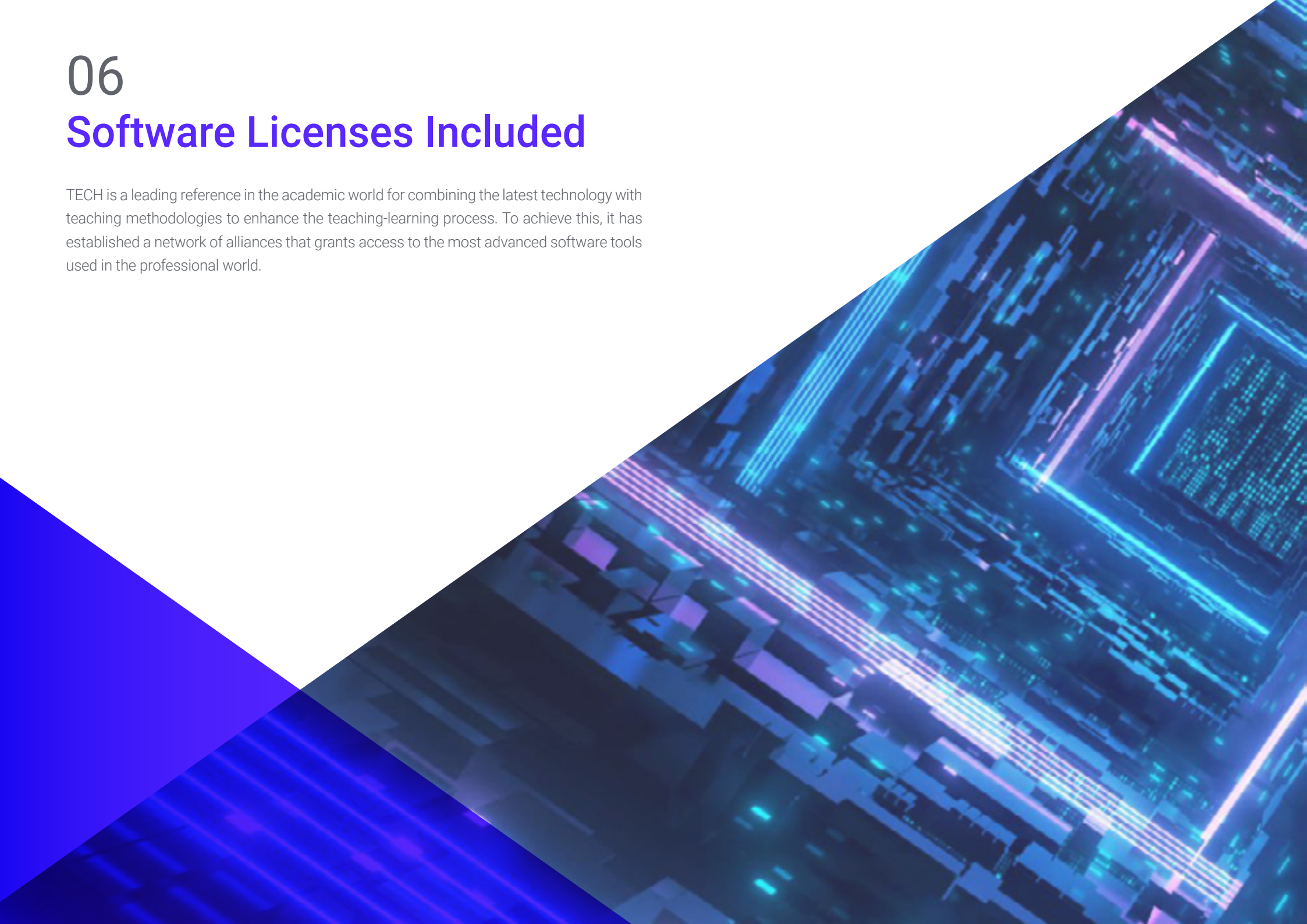
After completing the university program, you will be able to apply your knowledge and skills in the following positions:

- 1. Artificial Intelligence Engineer in software development:** Specialist in the creation and integration of AI algorithms in web and mobile environments, ensuring intelligent and scalable solutions.
- 2. Software Architect with AI:** Designer of advanced technology infrastructures that incorporate AI for system optimization and process automation.
- 3. Specialist in Machine Learning and Data Science:** Responsible for designing, training and optimizing Machine Learning models to solve complex problems in industrial, financial and technology sectors.
- 4. Chatbot and Virtual Assistant Developer:** Responsible for designing and optimizing intelligent assistants, integrating AI into customer service platforms, e-commerce and automated services.
- 5. AI Implementation Consultant:** Advisor to technology companies on the integration of Artificial Intelligence, ensuring the adoption of innovative tools to improve productivity and efficiency.
- 6. QA Testing Specialist with AI:** Leader in the automation of software quality tests, using AI tools to optimize development and improve the user experience.
- 7. Data Analyst and AI in Cloud Computing:** Expert in the management and processing of large volumes of data using Artificial Intelligence techniques in cloud environments.
- 8. Engineer in AI Security and Ethics:** Responsible for evaluating risks in the implementation of AI, guaranteeing privacy, data protection and regulatory compliance in technology projects.

06

Software Licenses Included

TECH is a leading reference in the academic world for combining the latest technology with teaching methodologies to enhance the teaching-learning process. To achieve this, it has established a network of alliances that grants access to the most advanced software tools used in the professional world.



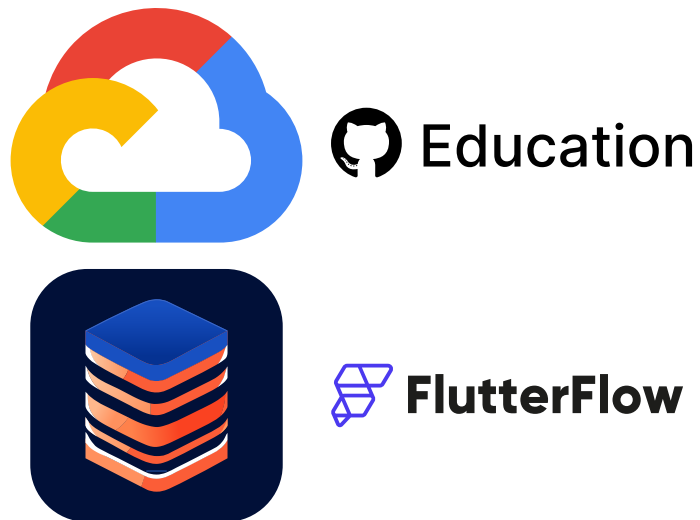
“

Upon enrolling, you will receive, completely free of charge, academic credentials for the following professional software applications”

TECH has established a network of professional alliances with the leading providers of software applied to various professional fields. These alliances allow TECH to access hundreds of software applications and licenses, making them available to its students.

The academic software licenses will allow students to use the most advanced applications in their professional field, so they can become familiar with them and master their use without incurring additional costs. TECH will handle the hiring process so that students can use them without limits during the time they are studying the Master's Degree in Artificial Intelligence in Programming, and they will be able to do so completely free of charge.

TECH will provide free access to the following software applications:



Google Career Launchpad

Google Career Launchpad is a solution for developing digital skills in technology and data analysis. With an estimated value of **5,000 dollars**, it is included **for free** in TECH's university program, providing access to interactive labs and certifications recognized in the industry.

This platform combines technical training with practical cases, using technologies such as BigQuery and Google AI. It offers simulated environments to work with real data, along with a network of experts for personalized guidance.

Github Education

GitHub Education is a collaborative platform for software development, offering professional tools for version control and project management. With an approximate value of **120 euros** annually, it is provided **free of charge** during the university program.

This platform promotes collaborative work in real-world scenarios, enabling users to manage repositories, collaborate in teams, and automate workflows. It includes GitHub Copilot, support for multiple IDEs, and access to technical resources that enhance development and code review capabilities in various languages and professional contexts.

AskYourDatabase

AskYourDatabase is an innovative solution for analyzing data using natural language, valued at **150 dollars**. During the program, graduates will use it **free of charge**, taking advantage of its ability to generate insights without writing code.

This platform promotes collaborative work in real-world scenarios, enabling users to manage repositories, collaborate in teams, and automate workflows. It includes GitHub Copilot, support for multiple IDEs, and access to technical resources that enhance development and code review capabilities in various languages and professional contexts.

Flutterflow

Flutterflow is an innovative platform based on Flutter that allows users to design, prototype, and create mobile and web applications in a visual and intuitive way, without the need to write code. Its simplified development environment significantly reduces production times and facilitates the creation of functional and professional digital solutions.

Throughout this university program, students will have **free access** to a full license of **Flutterflow**, valued at approximately **360 dollars**. This tool will provide them with the opportunity to gain direct experience in the development of real applications, strengthening key skills in interface design, business logic, and cross-platform deployment, without incurring additional costs.

07

Study Methodology

TECH is the world's first university to combine the **case study** methodology with **Relearning**, a 100% online learning system based on guided repetition.

This disruptive pedagogical strategy has been conceived to offer professionals the opportunity to update their knowledge and develop their skills in an intensive and rigorous way. A learning model that places students at the center of the educational process giving them the leading role, adapting to their needs and leaving aside more conventional methodologies.





TECH will prepare you to face new challenges in uncertain environments and achieve success in your career”

The student: the priority of all TECH programs

In TECH's study methodology, the student is the main protagonist.

The teaching tools of each program have been selected taking into account the demands of time, availability and academic rigor that, today, not only students demand but also the most competitive positions in the market.

With TECH's asynchronous educational model, it is students who choose the time they dedicate to study, how they decide to establish their routines, and all this from the comfort of the electronic device of their choice. The student will not have to participate in live classes, which in many cases they will not be able to attend. The learning activities will be done when it is convenient for them. They can always decide when and from where they want to study.

“

*At TECH you will NOT have live classes
(which you might not be able to attend)”*



The most comprehensive study plans at the international level

TECH is distinguished by offering the most complete academic itineraries on the university scene. This comprehensiveness is achieved through the creation of syllabi that not only cover the essential knowledge, but also the most recent innovations in each area.

By being constantly up to date, these programs allow students to keep up with market changes and acquire the skills most valued by employers. In this way, those who complete their studies at TECH receive a comprehensive education that provides them with a notable competitive advantage to further their careers.

And what's more, they will be able to do so from any device, pc, tablet or smartphone.

“*TECH's model is asynchronous, so it allows you to study with your pc, tablet or your smartphone wherever you want, whenever you want and for as long as you want*”

The case method has been the learning system most used by the world's best business schools. Developed in 1912 so that law students would not only learn the law based on theoretical content, its function was also to present them with real complex situations. In this way, they could make informed decisions and value judgments about how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

This action-oriented method will be applied throughout the entire academic itinerary that the student undertakes with TECH. Students will be confronted with multiple real-life situations and will have to integrate knowledge, research, discuss and defend their ideas and decisions. All this with the premise of answering the question of how they would act when facing specific events of complexity in their daily work.



Relearning Methodology

At TECH, case studies are enhanced with the best 100% online teaching method: Relearning.

This method breaks with traditional teaching techniques to put the student at the center of the equation, providing the best content in different formats. In this way, it manages to review and reiterate the key concepts of each subject and learn to apply them in a real context.

In the same line, and according to multiple scientific researches, reiteration is the best way to learn. For this reason, TECH offers between 8 and 16 repetitions of each key concept within the same lesson, presented in a different way, with the objective of ensuring that the knowledge is completely consolidated during the study process.

Relearning will allow you to learn with less effort and better performance, involving you more in your specialization, developing a critical mindset, defending arguments, and contrasting opinions: a direct equation to success.



A 100% online Virtual Campus with the best teaching resources

In order to apply its methodology effectively, TECH focuses on providing graduates with teaching materials in different formats: texts, interactive videos, illustrations and knowledge maps, among others. All of them are designed by qualified teachers who focus their work on combining real cases with the resolution of complex situations through simulation, the study of contexts applied to each professional career and learning based on repetition, through audios, presentations, animations, images, etc.

The latest scientific evidence in the field of Neuroscience points to the importance of taking into account the place and context where the content is accessed before starting a new learning process. Being able to adjust these variables in a personalized way helps people to remember and store knowledge in the hippocampus to retain it in the long term. This is a model called Neurocognitive context-dependent e-learning that is consciously applied in this university qualification.

In order to facilitate tutor-student contact as much as possible, you will have a wide range of communication possibilities, both in real time and delayed (internal messaging, telephone answering service, email contact with the technical secretary, chat and videoconferences).

Likewise, this very complete Virtual Campus will allow TECH students to organize their study schedules according to their personal availability or work obligations. In this way, they will have global control of the academic content and teaching tools, based on their fast-paced professional update.



The online study mode of this program will allow you to organize your time and learning pace, adapting it to your schedule”

The effectiveness of the method is justified by four fundamental achievements:

1. Students who follow this method not only achieve the assimilation of concepts, but also a development of their mental capacity, through exercises that assess real situations and the application of knowledge.
2. Learning is solidly translated into practical skills that allow the student to better integrate into the real world.
3. Ideas and concepts are understood more efficiently, given that the example situations are based on real-life.
4. Students like to feel that the effort they put into their studies is worthwhile. This then translates into a greater interest in learning and more time dedicated to working on the course.

The university methodology top-rated by its students

The results of this innovative teaching model can be seen in the overall satisfaction levels of TECH graduates.

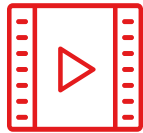
The students' assessment of the teaching quality, the quality of the materials, the structure of the program and its objectives is excellent. Not surprisingly, the institution became the top-rated university by its students according to the global score index, obtaining a 4.9 out of 5.

Access the study contents from any device with an Internet connection (computer, tablet, smartphone) thanks to the fact that TECH is at the forefront of technology and teaching.

You will be able to learn with the advantages that come with having access to simulated learning environments and the learning by observation approach, that is, Learning from an expert.



As such, the best educational materials, thoroughly prepared, will be available in this program:



Study Material

All teaching material is produced by the specialists who teach the course, specifically for the course, so that the teaching content is highly specific and precise.

This content is then adapted in an audiovisual format that will create our way of working online, with the latest techniques that allow us to offer you high quality in all of the material that we provide you with.



Practicing Skills and Abilities

You will carry out activities to develop specific competencies and skills in each thematic field. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop within the framework of the globalization we live in.



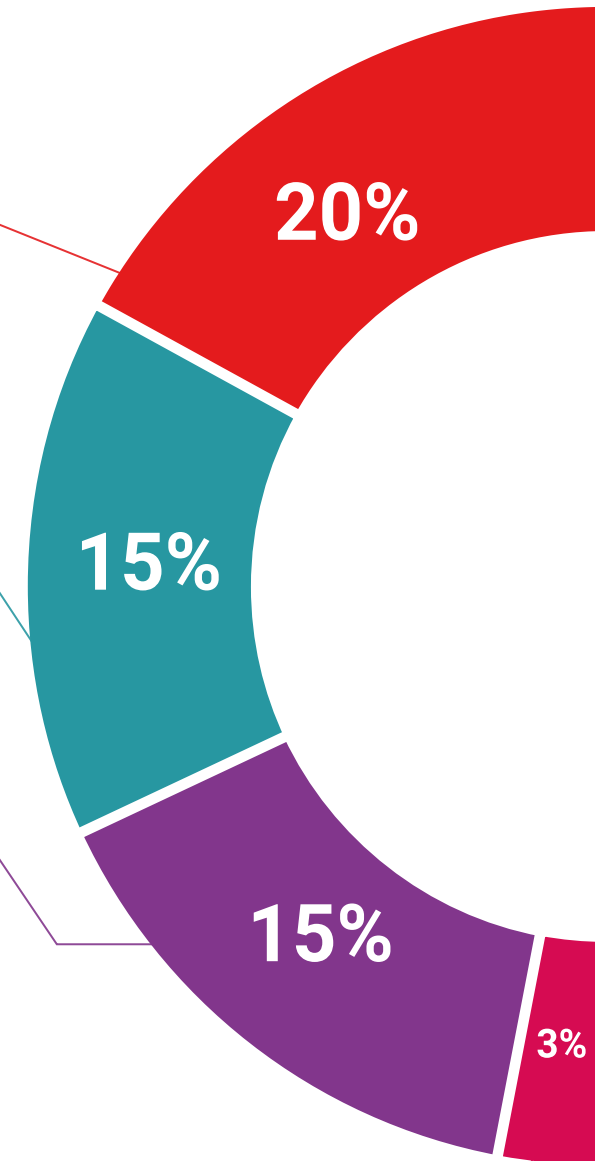
Interactive Summaries

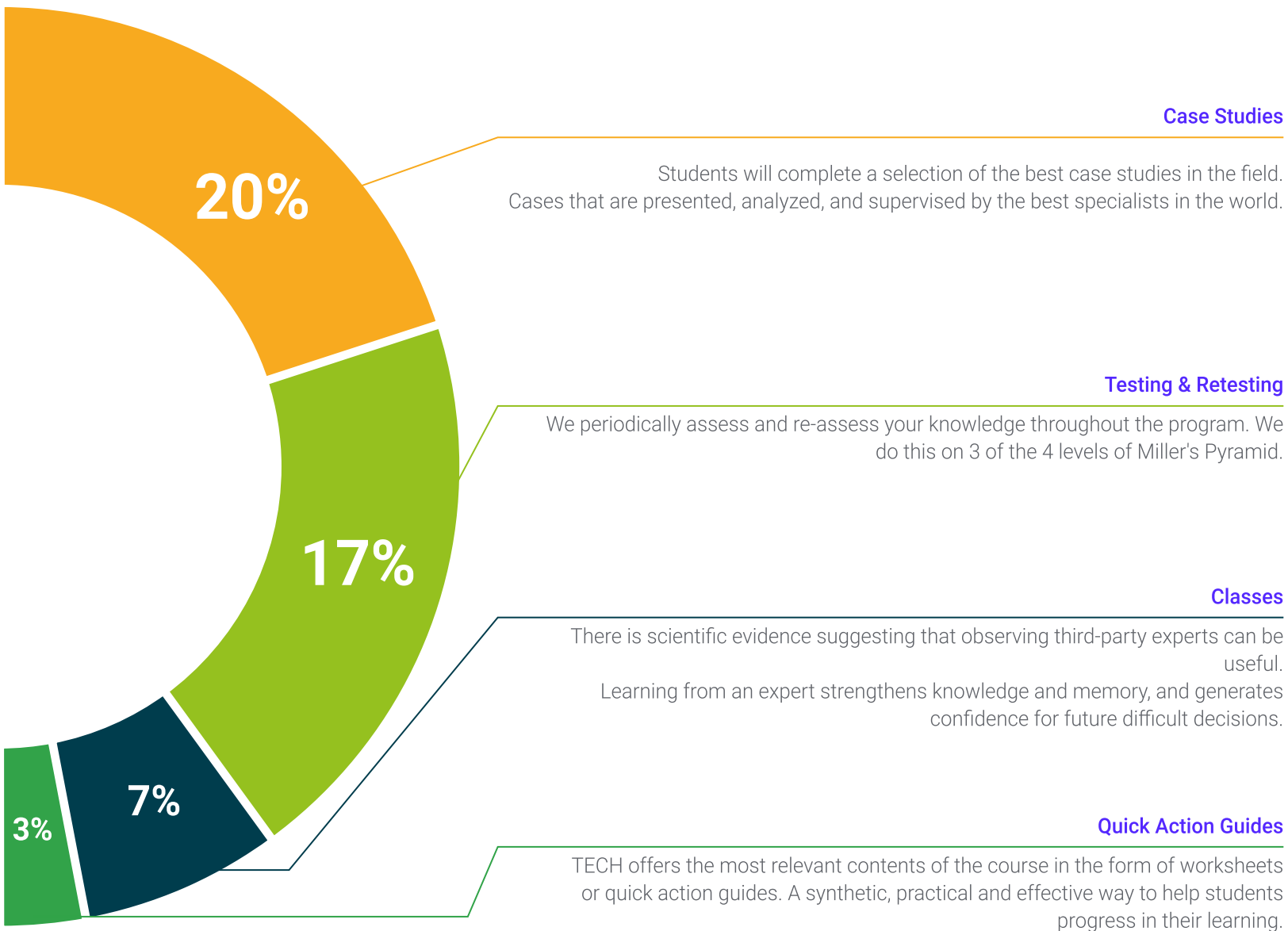
We present the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge. This exclusive educational system for presenting multimedia content was awarded by Microsoft as a "European Success Story".



Additional Reading

Recent articles, consensus documents, international guides... In our virtual library you will have access to everything you need to complete your education.





08

Teaching Staff

True to its commitment to offering the highest quality education, TECH has a top-level teaching staff. For this reason, the experts who make up this university degree stand out for their in-depth knowledge of AI in Programming, as well as having years of professional experience behind them. Thanks to these aspects, this academic pathway provides students with the best technology tools and strategies not only to broaden their understanding, but also to enable them to acquire skills to perfect their computer procedures. Students therefore have the guarantees they need to specialize in a digital sector that offers numerous opportunities.



“

The leading experts in Machine Learning in Programming have come together in this university degree to share with you all their knowledge in this sector”

Management



Dr. Peralta Martín-Palomino, Arturo

- CEO and CTO at Prometheus Global Solutions
- CTO at Korporate Technologies
- CTO at AI Shepherds GmbH
- Consultant and Strategic Business Advisor at Alliance Medical
- Director of Design and Development at DocPath
- Doctorate in Psychology from the University of Castilla La Mancha
- Doctorate in Economics, Business and Finance from the Camilo José Cela University
- Doctorate in Psychology from University of Castilla La Mancha
- Master's Degree in Executive MBA from the Isabel I University
- Master's Degree in Sales and Marketing Management from the Isabel I University
- Expert Master's Degree in Big Data by Hadoop Training
- Master's Degree in Advanced Information Technologies from the University of Castilla La Mancha
- Member of: SMILE Research Group



Mr. Castellanos Herreros, Ricardo

- Chief Technology Officer at OWQLO
- Specialist in Computer Systems Engineering and Machine Learning Engineer
- Freelance Technical Consultant
- Mobile Applications Developer for eDreams, Fnac, Air Europa, Bankia, Cetelem, Banco Santander, Santillana, Groupón and Grupo Planeta
- Web Developer for Openbank and Banco Santander
- Technical Engineer in Computer Systems from the University of Castilla la Mancha

09

Certificate

The Master's Degree in Artificial Intelligence in Programming guarantees students, in addition to the most rigorous and up-to-date education, access to a Postgraduate Certificate issued by TECH Global University..



“

Successfully complete this program and receive your university qualification without having to travel or fill out laborious paperwork"

This private qualification will allow you to obtain a **Master's Degree in Artificial Intelligence in Programming** endorsed by **TECH Global University**, the world's largest online university.

TECH Global University is an official European University publicly recognized by the Government of Andorra ([official bulletin](#)). Andorra is part of the European Higher Education Area (EHEA) since 2003. The EHEA is an initiative promoted by the European Union that aims to organize the international training framework and harmonize the higher education systems of the member countries of this space. The project promotes common values, the implementation of collaborative tools and strengthening its quality assurance mechanisms to enhance collaboration and mobility among students, researchers and academics.

This **TECH Global University** private qualification is a European program of continuing

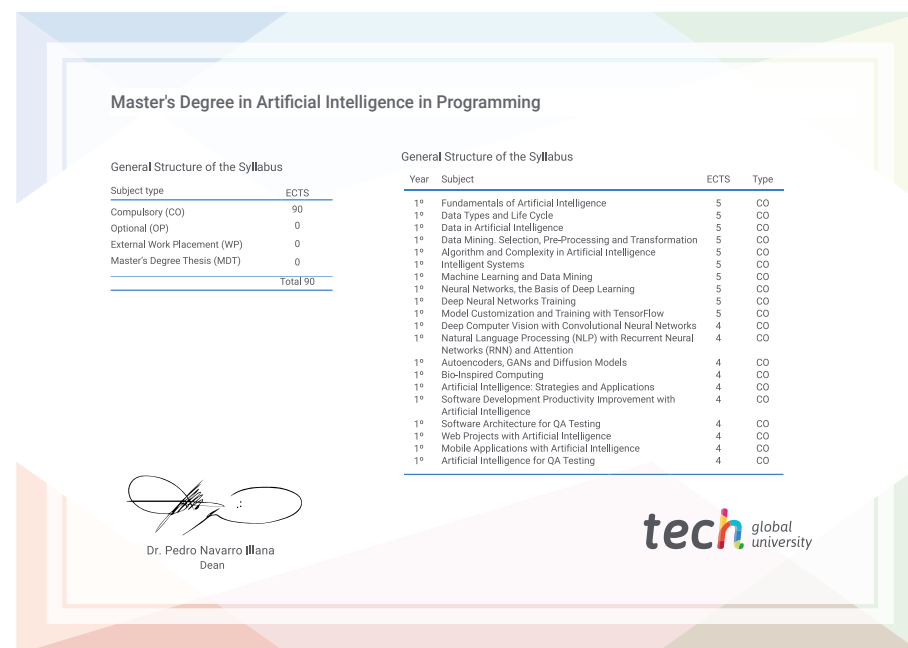
education and professional updating that guarantees the acquisition of competencies in its area of knowledge, providing a high curricular value to the student who completes the program.

Title: **Master's Degree in Artificial Intelligence in Programming**

Modality: **online**

Duration: **12 months**

Accreditation: **90 ECTS**





Master's Degree Artificial Intelligence in Programming

- » Modality: online
- » Duration: 12 months
- » Certificate: TECH Global University
- » Credits: 90 ECTS
- » Schedule: at your own pace
- » Exams: online

Master's Degree Artificial Intelligence in Programming