



UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL VILLA MARIA

CARRERA ACADÉMICA

Carrera	Ingeniería en Sistemas de Información		
Asignatura	INTELIGENCIA ARTIFICIAL	Nivel	V
Departamento	Sistemas de Información		
Plan de Estudios	2008	Régimen de cursado	cuatrimestral
	Carga horaria semanal (hs. Cátedras)		4
	Carga horaria total de la asignatura (hs. Cátedras)		96
Área	Tecnologías Básicas - Programación		
	%de horas cátedra del área en la carrera		
	%de horas cátedra de la asignatura en el área		
Ciclo Académico	2020		
Profesor Adjunto	Jorge Palombarini	J.T.P.	Juan Cruz Barsce

PLANIFICACIÓN DE LA ASIGNATURA

FUNDAMENTACIÓN DE LA ASIGNATURA

A partir de la evolución de los sistemas cognitivos y el aprendizaje profundo (*deeplearning*) [1][2], la ingeniería de sistemas de información ha ingresado definitivamente en un cambio paradigmático promovido por la creciente incorporación de la autonomía para actuar en el diseño y la programación de los sistemas de información, y su utilización como principal fuente de innovación y generación de valor en el desarrollo de software para distintas áreas. Como ejemplos representativos que sustentan la actualidad de esta aseveración, considérense el IBM® Watson™ [3], AlphaGo y AlphaGo Zero [4,5], OpenAI FIVE [6] y la arquitectura cognitiva Soar [7] para el desarrollo de aplicaciones (ver <http://www.soartech.com>) en ámbitos como el militar, gubernamental, educativo e industrial, o la reciente creación del Nodo de Colaboración Científico Industrial para la Investigación y el Desarrollo de la Inteligencia Artificial en la ciudad de Córdoba [8]. La mencionada autonomía impone que el diseño del sistema enfatice la capacidad intrínseca de aprender y, a partir de ello, actuar y decidir inteligentemente para hacer frente a las incertidumbres del entorno sobre el que actúa. La capacidad de aprender como rasgo distintivo del diseño y construcción de una entidad autónoma de software (**agente**) permite abordar tareas complejas sin necesidad de incluir a priori la lógica de actuación de manera explícita en la programación. Por el contrario, el sistema debe ser capaz de transformar una meta, rol, objetivo o requerimiento a satisfacer en una política de actuación.

A tal efecto, incorpora en su diseño una máquina de aprendizaje automático que le permite transformar la experiencia (real o simulada) en conocimiento y, sustentado en la capacidad aprender, controlar y adaptarse permanentemente al entorno que le influye y sobre el que actúa, potenciando las capacidades del operador a través de una simbiosis entre el usuario y el sistema que se logra por emulación de las capacidades humanas

para solución de problemas y toma de decisiones en forma reactiva, lo que conlleva significativos ahorros de tiempo y dinero junto a la posibilidad de generar/acumular conocimiento por medio de las sucesivas interacciones (Ver Figura 1).

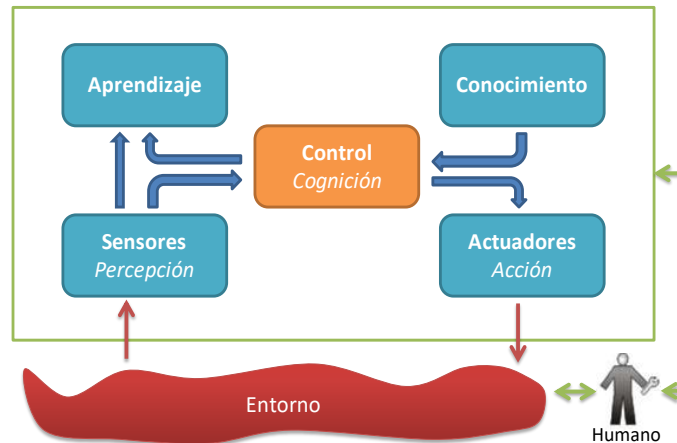


Figura 1. Arquitectura de un Sistema Cognitivo con lazo cerrado de percepción-acción

En consecuencia, la incorporación de las capacidades especificadas basadas en la utilización de técnicas de Inteligencia Artificial permite agregar valor a partir de convertir los sistemas tradicionales de software en sistemas que pueden razonar utilizando cantidades sustanciales de conocimiento representado apropiadamente, aprender de su experiencia de manera tal de llevar a cabo sus tareas con mejor performance mañana que hoy, poseer habilidades de auto-explicación y recepción de órdenes, ser conscientes de sus propias capacidades y reflejarlo en su comportamiento, y responder de manera robusta a la sorpresa, alcanzando altos niveles de confiabilidad, flexibilidad, adaptividad, y agilidad, facilitando al mismo tiempo la interacción y cooperación con humanos que poseen conocimiento experto sobre el dominio [6, 9].

El diseño, implementación e integración computacional de las mencionadas capacidades en un agente típicamente requiere el empleo de técnicas de *aprendizaje automático* o *machine learning* que se han convertido en el standard actual de facto para la generación de comportamiento y funcionalidad “inteligente” en sistemas software, proporcionando las bases fundamentales sobre las cuales se estructura el programa propuesto para la materia. De esta manera, las técnicas de aprendizaje se clasifican de manera amplia en tres categorías o ejes principales (ver Figura 2), dependiendo de la existencia de una señal de realimentación o bien si la tarea consiste o no en resolver un proceso de decisión en presencia de una señal de recompensa.

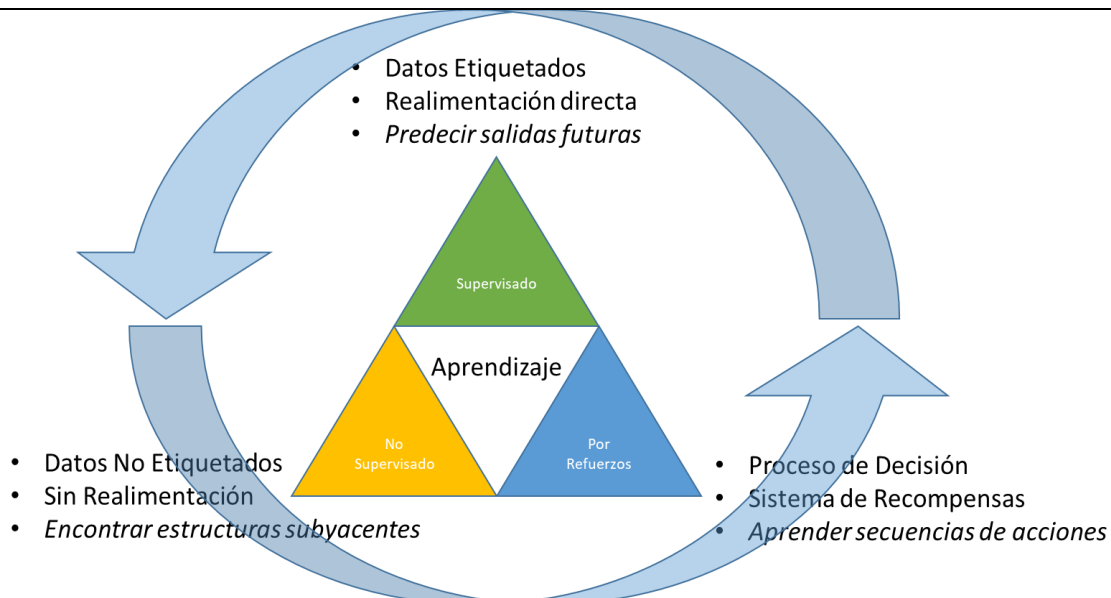


Figura 2. Tipos de técnicas de aprendizaje

En el caso del **aprendizaje supervisado** [10], se presentan al agente ejemplos formados por tuplas etiquetadas entrada-salida deseada dados por un “*teacher*” y el objetivo es generar una regla general que mapee entradas a salidas, aunque la entrada no haya sido vista durante el entrenamiento. En el **aprendizaje no supervisado** [10], los ejemplos no se encuentran etiquetados, y la tarea del agente consiste en encontrar algún tipo de estructura en la entrada. El aprendizaje no supervisado puede consistir en un objetivo en sí mismo (descubrir patrones ocultos en los datos) o un medio para lograr otro objetivo (aprendizaje de características o *features*). Finalmente, en el **aprendizaje por refuerzos** [11], los datos de entrenamiento (en forma de recompensas y castigos) se proporcionan como retroalimentación a las acciones del agente desde el entorno dinámico (real o simulado) sobre el que actúa. En este último caso, el objetivo del agente es desarrollar una política para controlar efectivamente su entorno, como es el caso de un agente que controla un vehículo o juega un juego contra un oponente.

En este contexto, en la actualidad la gran mayoría de los algoritmos que constituyen el estado del arte de la Inteligencia Artificial y que son empleados a nivel práctico para resolver problemas de predicción de valores, clasificación o desarrollo de políticas de actuación, se relaciona de manera directa con alguna de las tres áreas mencionadas anteriormente. Por caso, las aplicaciones de búsqueda en grandes cantidades de datos, interpretación de imágenes en medicina y agricultura, mapeo y ubicación en tiempo real, visión en drones y coches auto-piloteados emplean visión por computadora en procesos de clasificación, localización y detección de objetos. Los avances recientes en redes neuronales profundas (*deep learning*) y máquinas de soporte vectorial (*SVMs*) han producido mejoras sustanciales en el estado del arte y la performance de este tipo de sistemas. En otras áreas como el desarrollo de software autónomo para la resolución de problemas de decisión en tiempo real tales como juegos o ruteo automático, el uso de Aprendizaje por Refuerzos integrado con *deep learning* ha producido avances decisivos que se reflejan en la performance de niveles humanos o superiores en aplicaciones Atari, Dota 2, o Go [12,13]. De esa manera, se considera que las categorías o ejes mencionados deben servir como guía general y marco conceptual para el desarrollo de los temas del programa de la materia Inteligencia Artificial para la carrera de Ingeniería en Sistemas de Información. Teniendo esto en cuenta, se plantean los siguientes objetivos para la materia, y a continuación, los contenidos organizados por unidades, atento a los contenidos mínimos establecidos en la Ord. UTN 1150 y al diseño curricular de la carrera.

Referencias

- [1] Brachman R. Systems That Know What They're Doing. *IEEE Intelligent Systems*, 6, 67-71, 2002.
- [2] LeCun, Y., Bengio, Y., Hinton, G. Deep learning. *Nature*, 521, pp. 436-444, 2015.
- [3] High, R. The Era of Cognitive Systems: An Inside Look at IBM Watson and How it Works. IBM Red Guides for Business Leaders. www.redbooks.ibm.com/redpapers, 2013.
- [4] Silver, D. *Mastering the game of Go with deep neural networks and tree search*. *Nature* 529, p. 484-489, 2016.
- [5] Silver et al. *Mastering the game of Go without human knowledge*. *Nature* 550, p. 354- 359, 2017.
- [6] OpenAI Five. <https://blog.openai.com/openai-five/>. Accedida el 07/09/2018
- [7] Laird, J. E. *The Soar Cognitive Architecture*. MIT Press, Cambridge: MA, 2012.
- [8] Los tres lados de la inteligencia artificial fijan sede en Córdoba - <https://www.lanacion.com.ar/2169618-los-tres-lados-inteligencia-artificial-fijan-sede> Accedida el 07/09/2018
- [9] Van Otterlo, M. *The logic of adaptive behavior: Knowledge representation and algorithms for adaptive sequential decision making under uncertainty in first-order and relational domains*. IOS Press, Amsterdam, 2009.
- [10] James, G., Witten, J., Hastie, T., Tibshirani, R. *An Introduction to Statistical Learning with Applications in R*. Springer Science, New York, 2015.
- [11] Sutton R., Barto A. *Reinforcement Learning: An Introduction (2nd ed.)*. MIT Press, 2018.
- [12] Mnih et al. *Human-level control through Deep Reinforcement Learning*. *Nature* 518, p. 529-533, 2015.
- [13] Li, Y. *Deep Reinforcement Learning: An Overview*. arXiv preprint: arXiv:1701.07274v5 [cs.LG], 2017.

OBJETIVOS

1. COGNITIVOS: CONOCER Y COMPRENDER

- Conocer y comprender los problemas que aborda la Inteligencia Artificial en general, y los Sistemas Inteligentes en particular.
- Conocer y comprender los agentes inteligentes y su diseño, sus distintos tipos, los ambientes en donde deben desenvolverse y la aplicabilidad en distintas situaciones planteadas.
- Conocer y comprender las bases fundamentales del aprendizaje supervisado, no supervisado y por refuerzos.
- Conocer y comprender las técnicas básicas de resolución de problemas de Inteligencia Artificial derivadas de los tres tipos de aprendizaje, y su integración en sistemas de información.
- Conocer y comprender la aplicabilidad, el desarrollo y la arquitectura de los sistemas inteligentes artificiales.

2. PROCEDIMENTALES: BRINDAR APTITUDES PARA

- Modelar, implementar y testear algoritmos basados en técnicas de Inteligencia Artificial.
- Integrar los algoritmos desarrollados en prototipos de aplicaciones software.
- Implementar Sistemas Inteligentes utilizando lenguajes y herramientas de Inteligencia Artificial.
- Intervenir en el desarrollo de sistemas Cognitivos, Basados en Conocimiento y Sistemas Expertos.

3. ACTITUDINALES: PROMOVER DISPOSICIONES PARA

- Aplicar las herramientas de la Inteligencia Artificial a la resolución de problemas prácticos.

- Actuar inter y multidisciplinariamente, con enfoque Sistémico.
- Aplicar metodologías de representación y resolución de problemas utilizadas en Ingeniería Artificial y emplearlas en el abordaje de situaciones que se presentarán en la actividad profesional.
- Adquirir habilidades para el trabajo en equipo y entre equipos mediante la integración de componentes basados en Inteligencia Artificial a software en desarrollo o ya desarrollados por otras entidades para agregar valor.
- Adquirir valores éticos en la aplicación de la Inteligencia Artificial a problemas sensibles para la sociedad.

CONTENIDOS

UNIDAD 1 Introducción a la Inteligencia Artificial. Definiciones e historia de la Inteligencia Artificial. Áreas de aplicación. Concepto de Agente Inteligente y Agente Cognitivo. Herramientas y técnicas provistas por la IA: una breve reseña. Otros enfoques y tecnologías empleados como soportes en la toma de decisiones.

Bibliografía

Inteligencia Artificial: Un Enfoque Moderno, Cap. 1, 2.

UNIDAD 2 Resolución de problemas mediante búsqueda. Búsqueda de soluciones. Estrategias para búsqueda no informada. Búsqueda Heurística. Funciones Heurísticas. Evaluación de Complejidad.

Bibliografía

Inteligencia Artificial: Un Enfoque Moderno, Cap. 3.

UNIDAD 3 Introducción al Aprendizaje Supervisado y No Supervisado. Regresión y Clasificación. K vecinos más próximos. Equilibrio entre Varianza y Sesgo. Validación cruzada. Curvas ROC. Aprendizaje no Supervisado. Análisis de Componentes Principales. Clustering.

Bibliografía

An Introduction to Statistical Learning. Caps. 2, 3, 4, 5, 7, 10.

Information Theory, Inference, and Learning Algorithms. Cap. 4.

Deep Learning, Cap. 5, 13.

Machine Learning: A probabilistic Perspective. Cap. 1.

All of Statistics: A concise course in Statistical inference. Cap. 24.

UNIDAD 4 Aprendizaje Estadístico. Inferencia Estadística Clásica. Inferencia Bayesiana. Regresión Logística. Clasificador Naive Bayes. Introducción a la Minería de Texto. Máquinas de Soporte Vectorial. Métodos basados en Árboles.

Bibliografía

An Introduction to Statistical Learning. Caps. 8, 9.

Information Theory, Inference, and Learning Algorithms. Cap. 4.

Machine Learning: A probabilistic perspective. Cap. 3.



All of Statistics: A concise course in Statistical inference. Cap. 12, 24.

Inteligencia Artificial: Un Enfoque Moderno, Cap. 18.

UNIDAD 5 Introducción al Aprendizaje por refuerzos. Elementos del Aprendizaje por refuerzos. Retroalimentación Evaluativa. Métodos *model free* y *model based*. Algoritmos de aprendizaje. Integración de Modelos y Trazas. Generalización y Abstracción de conocimiento.

Bibliografía

Reinforcement Learning: An Introduction: Cap. 1, 3, 4, 5, 6, 8, 12.

UNIDAD 6 Algoritmos Genéticos. Introducción. Elementos: Individuos, Genes, Población, Aptitud. Codificación. Operadores: Selección, Recombinación, Mutación, Reemplazo. Criterios de Convergencia. Problemas de aplicación.

Bibliografía

Introduction to Genetic Algorithms, Cap. 1, 2, 3.

UNIDAD 7 Redes Neuronales. Origen del modelo neuronal. Componente elemental del modelo. Modelo "Perceptron". Redes de capas múltiples. Algoritmos de aprendizaje. Deep Learning: Deep Feedforward Networks. Aprendizaje basado en el gradiente. Unidades Ocultas. Diseño de Arquitectura. Backpropagation. Redes Convolucionales. Operación de Convolución. Pooling. Salidas estructuradas.

Bibliografía

Neural Networks A Comprehensive Foundation, Cap. 1, 2, 3, 4, 5.

Deep Learning, Cap. 6, 7, 9.

UNIDAD 8 Representación de Conocimiento. El problema de la representación de conocimiento. Paradigmas. Formalismos lógicos. Cálculo de proposiciones. Cálculo de predicados. Automatización de la Lógica. Esquemas de representación. Fundamentos de Sistemas Cognitivos y Arquitecturas Cognitivas. Planificación: Representación de conocimiento para planeamiento. Operadores de planificación en el marco de la arquitectura Soar. Descripción de estados y objetivos. Métodos "Forward/Backward Search". Modelo Computacional Espacio Problema. Búsqueda y Adquisición de Conocimiento. Impasses y Subestados. Ciclo de Procesamiento de Soar. Chunking y RL en Soar.

Bibliografía

Inteligencia Artificial: Un Enfoque Moderno, Caps. 7, 8, 9, 10, 11, 12.

The Soar Cognitive Architecture, Caps. 1, 3, 4, 5, 6, 7.

UNIDAD 9 Redes semánticas y Marcos. Razonamiento Basado en Casos. Lenguajes empleados en la representación de conocimiento. Razonamiento bajo incertidumbre. Razonamiento Probabilístico. Representación de Conocimiento en entornos inciertos. Redes Bayesianas. Inferencia en Redes Bayesianas. Modelos Probabilísticos Relacionales y de Primer Orden.

Bibliografía

Inteligencia Artificial: Un Enfoque Moderno, Cap. 13, 14, 15.

ESTRATEGIAS METODOLÓGICAS

a) **Estrategias de enseñanza:** La base de la propuesta metodológica planteada consiste en el dictado de clases magistrales haciendo uso de mapas conceptuales para representar esquemáticamente los conceptos y las relaciones de los temas concretos a dictar en la clase. El objetivo es organizar y presentar visualmente los contenidos de la clase teniendo en cuenta distinto niveles de abstracción y jerarquización, contextualizando la misma en el marco del programa y de los tres ejes principales: aprendizaje supervisado, no supervisado o por refuerzos. Además, la utilización de mapas conceptuales favorecerá la relación por parte de los alumnos los nuevos conceptos con aquellos que ya poseen y han adquirido en materias de área, como Matemática Superior, Simulación e Investigación Operativa. Para favorecer la interacción con los alumnos y su participación activa durante el desarrollo de la clase, y teniendo en cuenta el perfil del graduado, el dictado de la misma proseguirá empleando como sub-estrategia una integración de aprendizaje basado en problemas con estudio de caso; el objetivo principal de dicha integración es involucrar a los alumnos de modo activo en el aprendizaje de conocimientos y habilidades relacionadas con los contenidos, a través del planteamiento de un problema o situación compleja. En primer lugar, se definirá un problema o situación por parte del docente, a partir del cual se les solicitará a los estudiantes que aborden las diferentes fases que implica el proceso de resolución del problema o situación. Este tipo de metodología tiene el objetivo secundario de relacionar el problema empleado como disparador en el aspecto teórico, con el aprendizaje basado en proyectos de la parte práctica, ya que en ambos casos se atacará el mismo tipo de problema. El problema planteado se encontrará relacionado con un caso concreto de aplicación de la técnica que se pretende tratar en la clase correspondiente; así, mediante la descripción de una situación real, se pretende acercar a los estudiantes desde la teoría a la realidad. El caso empleado como ejemplo describirá un escenario global en el cual se conjugan diversas variables y que es susceptible de ser objeto de estudio para ser resuelto empleando la técnica en cuestión. Por ejemplo, se estudiará un caso en el cuál es necesario entrenar un modelo para determinar si un cliente de una tarjeta de crédito pagará o no el resumen de su tarjeta, en función de su historial de pagos y consumos (aprendizaje supervisado). Si la técnica objeto de la clase es Máquinas de Soporte Vectorial para clasificación, junto a los estudiantes se analizará el caso en cuestión, se definirán los problemas particulares que es necesario resolver antes del entrenamiento del modelo (ej. Tratamiento del dataset, análisis estadístico, etc.), y, finalmente, se procederá a explicar los aspectos teóricos involucrados en la técnica particular (hiperplano de margen máximo) y aspectos relacionados con su implementación en entornos de programación comúnmente empleados para el desarrollo y aplicación de técnicas de IA.

El desarrollo de las clases siguiendo la metodología propuesta se apoyará en el uso de herramientas TICS como proyector, internet (para el caso en que sea necesaria la visualización de videos en tiempo real) e IDEs tales como Pycharm para código Python.

b) **Modalidad de agrupamientos:** Los trabajos prácticos propuestos serán resueltos de manera individual o en grupos de hasta 2 alumnos por grupo, en los TPs habilitados a tal efecto.

c) **Consultas:** Las consultas serán individuales o grupales según el tipo de inquietud y se realizarán en horario de clases. Fuera del horario de clases, se podrán realizar consultas presenciales los días lunes de 10 a 12 hs. Si el alumno lo requiere, podrá realizar consultas específicas personalmente en otro día y horario pre pactado con los docentes. También podrá realizar consultas por correo electrónico dirigido a los docentes, o bien a través del espacio brindado a tal efecto en el Campus Virtual de la materia.

Consultas fuera del ámbito de Clase: Lunes de 10:00 a 12:00 hs.

Etapas del proceso: durante todo el cursado y previamente a los parciales, recuperatorios y/o exámenes finales.

d) **Organización de espacios dentro y fuera del ámbito universitario:**

Las clases se desarrollarán en las aulas y laboratorio de computación.

La cátedra cuenta con un repositorio en Github donde se pone a disposición de los alumnos toda la bibliografía recomendada (Obligatoria y Opcional) en formato digital, el programa y el contenido de las clases teóricas y prácticas. La dirección del repositorio es: <https://github.com/inteligenciafrvm/inteligenciafrvm>

En el caso particular de los Trabajos Prácticos, se realizará la gestión de entrega y correcciones a través del campus virtual de la UTN FRVM.

Se deberá utilizar Internet como fuente extra de investigación de las distintas temáticas propuesta.

Materiales curriculares (recursos):

- CLASES TEÓRICAS, PRÁCTICAS Y TEÓRICO- PRÁCTICAS
LABORATORIO DE COMPUTACIÓN para prácticos de desarrollo de software.
- BIBLIOGRAFÍA: libros, wikipedia, buscadores de la web, apuntes redactados por la cátedra, papers relacionados con los temas dictados, filminas del dictado de la materia.
- SOFTWARE: Python (numpy, sci-kit learn), Java, VB.NET o C#, Soar.

FORMACIÓN PRÁCTICA

a) Formación experimental

Ámbito de realización: Aulas de clases prácticas y en laboratorio de computación.

Tiempo (carga horaria): 20 horas cátedras en total.

Disponibilidad de infraestructura y equipamiento: Laboratorio de computación de la UTN.

Actividades a desarrollar:

- Resolución de guías de trabajos prácticos relacionadas con la temática

planteada.

- Planteo de problemas.
- Explicaciones grupales y personalizadas.

Evaluación (de seguimiento y final): 1 parcial teórico y 7 trabajos prácticos para la regularidad, y examen final teórico-práctico.

b) **Resolución de problemas de ingeniería**

Ámbito de realización: Aulas de clases prácticas y en laboratorio de computación

Tiempo (carga horaria): 20 horas cátedras en total.

Actividades a desarrollar:

- Resolución de guías de trabajos prácticos de problemas de ingeniería utilizando las técnicas vistas en clase.
- Programación de prototipos software para resolver problemas planteados, que incorporen a partir de su diseño un mecanismo de aprendizaje, empleando preferentemente Python, o en su defecto alguno de los lenguajes mencionados anteriormente.
- Planteo de problemas.
- Explicaciones grupales y personalizadas.

Evaluación (de seguimiento y final): 1 parcial escrito teórico y trabajos prácticos de resolución obligatoria, para la regularidad. Examen final teórico-práctico.

EVALUACIÓN

Momentos: Seguimiento Continuo de Desarrollo de TPs.

Un examen parcial teórico, con opción a un recuperatorio.

Instrumentos: Se llevarán registros de seguimiento de entregas y evaluación de informes de los trabajos prácticos, y de las notas alcanzadas. Examen parcial.

Actividades: Resolución de problemas y casos de aplicación en Laboratorio. Participación en clases teóricas y prácticas. Realización de prácticas. Presentación de trabajos prácticos acompañados del código fuente correspondiente.

Criterios de:

- A) **Regularidad:** Evaluación Continua, mediante Guías de Trabajos Prácticos, en Laboratorio de Computación. Se accederá a la regularidad mediante la aprobación del 60% de los trabajos prácticos propuestos por la Cátedra. La aprobación de cada práctico se obtendrá sacando al menos 60 puntos sobre 100. Se debe además cumplir con un 75% de la asistencia a las clases teórico-prácticas.
- B) **Aprobación Directa:** El alumno que cumpla las condiciones establecidas para la regularidad, accederá a la aprobación directa de la materia aprobando el parcial teórico con nota igual o superior a 80 puntos sobre 100, y con un promedio entre las notas de los TPs igual o superior a 80 puntos sobre 100.

Asignaturas o conocimientos con que se vincula:



UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL VILLA MARIA

Se apoya en Lógica, Simulación, Probabilidad y Estadística, Paradigmas de Programación, Álgebra, Lenguajes Formales, Diseño de Sistemas y Matemática Superior.

Actividades de coordinación:

El equipo de la cátedra coordina las tareas docentes como son: dictado de clases, confección de guías de TP, evaluaciones (parciales y finales), etc.

Cronograma:

Cl.	Día	Fecha	Tema
1	Miércoles	25/3/2020	Introducción a la Inteligencia Artificial. Definiciones e historia de la Inteligencia Artificial. Áreas de aplicación. Concepto de Agente Inteligente y Agente Cognitivo. Herramientas y técnicas provistas por la IA: una breve reseña. Otros enfoques y tecnologías empleados como soportes en la toma de decisiones.
2	Viernes	27/3/2020	Presentación de la parte práctica
3	Miércoles	1/4/2020	Resolución de problemas mediante búsqueda. Búsqueda de soluciones. Estrategias para búsqueda no informada. Búsqueda Heurística. Funciones Heurísticas. Evaluación de Complejidad.
4	Viernes	3/4/2020	Debate sobre definiciones de IA en cultura popular. Trabajar TP1
5	Miércoles	8/4/2020	Introducción al Aprendizaje Supervisado y No Supervisado. Regresión y Clasificación. K vecinos más próximos. Equilibrio entre Varianza y Sesgo.
6	Viernes	10/4/2020	FERIADO NACIONAL
7	Miércoles	15/4/2020	Validación cruzada. Curvas ROC. Aprendizaje no Supervisado. Análisis de Componentes Principales. Clustering.
8	Viernes	17/4/2020	Presentación y trabajo sobre TP1
9	Miércoles	22/4/2020	Aprendizaje Estadístico. Inferencia Estadística Clásica. Inferencia Bayesiana. Regresión Logística. Clasificador Naïve Bayes.
10	Viernes	24/4/2020	Trabajar TP1
11	Miércoles	29/4/2020	Introducción a la Minería de Texto. Máquinas de Soporte Vectorial. Métodos basados en Árboles
12	Viernes	1/5/2020	FERIADO NACIONAL
13	Miércoles	6/5/2020	Introducción al Aprendizaje por refuerzos. Elementos del Aprendizaje por refuerzos. Retroalimentación Evaluativa. Métodos model free y model based.
14	Viernes	8/5/2020	Presentación y trabajo sobre TP2
15	Miércoles	13/5/2020	Algoritmos de aprendizaje. Integración de Modelos y Trazas. Generalización y Abstracción de conocimiento.
16	Viernes	15/5/2020	Trabajar TP2

17	Miércoles	20/5/2020	Algoritmos de aprendizaje. Integración de Modelos y Trazas. Generalización y Abstracción de conocimiento.
18	Viernes	22/5/2020	Trabajar TP2
19	Miércoles	27/5/2020	Algoritmos Genéticos. Introducción. Elementos: Individuos, Genes, Población, Aptitud. Codificación. Operadores: Selección, Recombinación, Mutación, Reemplazo. Criterios de Convergencia. Problemas de aplicación.
20	Viernes	29/5/2020	Trabajar TP2
21	Miércoles	3/6/2020	Redes Neuronales. Origen del modelo neuronal. Componente elemental del modelo. Modelo "Perceptron". Redes de capas múltiples. Algoritmos de aprendizaje. Deep Learning: Deep Feedforward Networks.
22	Viernes	5/6/2020	Presentación y trabajo sobre TP3
23	Miércoles	10/6/2020	Aprendizaje basado en el gradiente. Unidades Ocultas. Diseño de Arquitectura. Backpropagation. Redes Convolucionales. Operación de Convolución. Pooling. Salidas estructuradas.
24	Viernes	12/6/2020	Trabajar TP3
25	Miércoles	17/6/2020	Representación de Conocimiento. El problema de la representación de conocimiento. Paradigmas. Formalismos lógicos. Cálculo de proposiciones. Cálculo de predicados. Automatización de la Lógica. Esquemas de representación. Fundamentos de Sistemas Cognitivos y Arquitecturas Cognitivas. Planificación: Representación de conocimiento para planeamiento. Operadores de planificación en el marco de la arquitectura Soar.
26	Viernes	19/6/2020	Trabajar TP3
27	Miércoles	24/6/2020	Descripción de estados y objetivos. Métodos "Forward/Backward Search". Modelo Computacional Espacio Problema. Búsqueda y Adquisición de Conocimiento. Impasses y Subestados. Ciclo de Procesamiento de Soar. Chunking y RL en Soar.
28	Viernes	26/6/2020	Trabajar TP3
29	Miércoles	1/7/2020	Redes semánticas y Marcos. Razonamiento Basado en Casos. Lenguajes empleados en la representación de conocimiento. Razonamiento bajo incertidumbre. Razonamiento Probabilístico. Representación de Conocimiento en entornos inciertos. Redes Bayesianas. Inferencia en Redes Bayesianas. Modelos Probabilísticos Relacionales y de Primer Orden.
30	Viernes	3/7/2020	Parcial Teórico
31	Miércoles	8/7/2020	Recuperatorio Teórico
32	Viernes	10/7/2020	FERIADO

Bibliografía:

a) Obligatoria o básica:



1. **Inteligencia Artificial. Un enfoque moderno.** S. Russell, P. Norvig. 3ra ed. Prentice Hall (2010).
2. **Reinforcement Learning: An Introduction** R. Sutton, A. Barto. 2nd. ed. MIT Press (2018).
3. **Introduction to Genetic Algorithms.** S. Sivanandam, S. Deepa. Springer (2008).
4. **An Introduction to Statistical Learning.** G. James, D. Witten, T. Hastie, R. Tibshirani, 2nd. Ed., Springer (2013).
5. **The Soar Cognitive Architecture,** J. Laird, MIT Press (2012).
6. **Neural Networks A Comprehensive Foundation.** S. Haykin, 2nd. ed, Prentice Hall (2005).

b) Complementaria:

1. **Machine Learning: A Probabilistic Perspective.** K. Murphy, MIT Press (2012).
2. **Information Theory, Inference, and Learning Algorithms.** D. McKay, Cambridge University Press (2005).
3. **Deep Learning.** I. Goodfellow, J. Bengio, A. Courville. MIT Press (2016).
4. **All of Statistics: A concise course in Statistical inference,** L. Wasserman. Springer (2010).
5. **Redes Neuronales y Sistemas Borrosos.** 2ª edición. Martin del Brio, Sanz Molina. Alfaomega (2005).
6. **Inteligencia Artificial. Segunda edición.** E, Rich. K. Knight. Mc Graw-Hill (1994).
7. **Numerical Recipes.** W. Press, S. Teukolsky, W. Vetterling, B. Flannery, 3rd Edition (2007)
8. **Minería de datos basada en Sistemas Inteligentes.** Britos, Hossian, García Martínez, Sierra. Nueva Librería (2005).

Distribución de tareas del equipo docente:

Coordinación entre los docentes Palombarini y Barsce.

Articulación docencia-investigación-extensión:

Los casos de estudio presentados se encuentran articulados de manera directa con los resultados de los proyectos de I+D en curso en los cuáles participan los docentes de la cátedra, a saber:

- Proyecto PID UTN PAUTIFE0007699TC “Rescheduling en línea y control de procesos industriales integrando aprendizaje por refuerzos con representaciones profundas basadas en grafos y analítica visual”. Director: Dr. Ernesto C. Martínez.

Por otra parte, los casos de estudio también se relacionan con los servicios tecnológicos prestados por los docentes de la cátedra como parte de su actividad de transferencia de conocimiento y tecnología al medio productivo a través de Servicios Tecnológicos de Alto Nivel (STAN) de CONICET, en concreto:

- Servicio Tecnológico de Alto Nivel ST3286 CONICET, “*Consultoría especializada para la integración de capacidades de Text Mining en plataformas online para soporte al cliente*”