



Propuesta de Trabajo Profesional  
Licenciatura en Análisis de Sistemas, Ingeniería en  
Informática

---

Sistema de detección en tiempo real de  
publicidad en la vía pública

---

*Tutor:* Ing. Martín Buchwald

del Mazo, Federico  
100029

Pastine, Casimiro  
100017

# Índice

<b>1. Motivación</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
<b>3. Descripción del problema</b>	<b>2</b>
3.1. Computer Vision . . . . .	2
3.2. Despliegue . . . . .	4
3.3. Tecnologías Relevantes al Trabajo . . . . .	4
<b>4. Esquema de Trabajo</b>	<b>5</b>
4.1. Metas Parciales . . . . .	5
4.2. Cronograma Final . . . . .	6
<b>Referencias</b>	<b>6</b>
<b>Bibliografía</b>	<b>6</b>

## 1. Motivación

Dentro del marco de trabajo profesional de las carreras de Licenciatura en Análisis de Sistemas y de Ingeniería en Informática de la Facultad de Ingeniería, Universidad de Buenos Aires, se presenta la siguiente propuesta de proyecto con el objetivo de desarrollar un sistema de detección de publicidades en la vía pública en tiempo real.

El interés detrás de este proyecto radica en la creciente intrusión e invasión de publicidades no deseadas en nuestro entorno cotidiano. Este proyecto busca concientizar y ofrecer una solución práctica a esta problemática, enfocándose principalmente en la detección, y luego permitiendo al usuario aplicar un post-procesamiento configurable para difuminar, resaltar o transformar de otra manera lo detectado.

## 2. Objetivos

- Aplicar los conocimientos interdisciplinarios de comprensión y evaluación de sistemas aprendidos en nuestra carrera académica en un campo enteramente nuevo para nosotros como lo es *Computer Vision*.
- Realizar un proyecto integral con su debida investigación y comprensión del estado del arte en las disciplinas relevantes, abarcando desde el entrenamiento de un modelo de clasificación de objetos hasta el despliegue del sistema en algún entorno fácilmente accesible por el usuario final.
- Encontrarnos con desafíos prácticos como la falta de los datos necesarios para el modelo a entrenar y la evaluación y comparación de distintos métodos para lograr un rendimiento aceptable y encontrar las soluciones adecuadas para los obstáculos que vayan apareciendo, sopesando las distintas alternativas posibles que surgirán durante el desarrollo del trabajo.
- Lograr hacer un sistema de código abierto, modularizable y extensible tal que en un futuro se pueda cambiar cualquier eslabón de su cadena como los objetos a detectar, el algoritmo de detección y rastreo de objetos o el post procesamiento a aplicar.

## 3. Descripción del problema

### 3.1. Computer Vision

El campo de *Computer Vision* se refiere a la disciplina de la inteligencia artificial que se ocupa de permitir a las computadoras “ver” y comprender el mundo visual de la misma manera que lo hacen los seres humanos. Utiliza diversas técnicas y algoritmos para adquirir, procesar, analizar y comprender imágenes y videos digitales con el objetivo de extraer información significativa y tomar decisiones basadas en ella.

En los últimos años este campo ha experimentado un crecimiento y una atención considerable debido al aumento del poder computacional, la disponibilidad de grandes conjuntos de datos y los avances en el aprendizaje automático y las redes neuronales.

Uno de los avances más revolucionarios dentro del campo de *Computer Vision* es el uso de las *Redes Neuronales Convolucionales (CNN)*. Son un tipo de arquitectura de redes neuronales especialmente diseñadas para el procesamiento de datos de imágenes o señales de audio. La característica determinante es la capa convolucional de la red, a través de la cual las CNN aprenden a extraer características y combinan esta información para realizar tareas de clasificación y reconocimiento de patrones.

Uno de los problemas iniciales de *Computer Vision* es la clasificación de imágenes: Partiendo de una imagen, queremos saber si la misma contiene un objeto deseado. Por ejemplo, podríamos tener una foto en la vía pública y querer saber si contiene o no a un cartel publicitario.



El paso siguiente a esto es la detección de objetos. Ya no solo importa si el cartel publicitario está presente, sino exactamente en qué porción de la imagen lo está.



Naturalmente el paso siguiente a detectar objetos en imágenes estáticas (fotografías), es hacerlo en video. Queremos detectar cuándo un objeto entra en escena, cuando sale, y poder seguirle el rastro en todo momento. Esto se denomina *Object Tracking* y es la técnica principal sobre la cual basamos nuestro trabajo.

El paso final en cuanto a la detección de objetos en nuestro trabajo incluye la meta de hacerlo en tiempo real. A diferencia de analizar un video pregrabado, donde es prescindible la velocidad de procesamiento, esto plantea el desafío considerable de tener que procesar los cuadros del video más rápido de lo que están siendo capturados para poder mostrar el resultado final inmediatamente.

### 3.2. Despliegue

En el despliegue del sistema se nos presentan alternativas que van a ser evaluadas en términos del rendimiento y la experiencia de usuario que buscamos alcanzar.

El mínimo entregable que proponemos para nuestro trabajo es una herramienta de procesamiento de video pregrabado que se corra localmente por línea de comando y devuelva un video procesado como salida.

Nuestra meta principal es entregar una experiencia de usuario con una interfaz mejor a la de línea de comando, que pueda ser utilizada desde cualquier parte, sin requerir ningún tipo de instalación previa. Por esto es que consideramos que la interfaz ideal del sistema es en un sitio web. El hecho de que la interfaz sea una página web nos da dos posibles arquitecturas a evaluar. Por un lado contemplamos la arquitectura cliente/servidor, mientras que su contraposición natural es un sistema que sea ejecutado solamente en el cliente local.

En una arquitectura cliente/servidor, la capacidad computacional del cliente no es relevante, pero la complejidad principal del sistema recae en poder minimizar la latencia entre el cliente y el servidor. Por otro lado, en una arquitectura sin servidor donde todo se ejecuta localmente, el principal desafío es poder seguir ofreciendo una buena experiencia de usuario independientemente del poder computacional del dispositivo sobre el cual corre el sistema. Sobre esta última idea, una alternativa disponible es la de utilizar *WebAssembly*, ya que nos permitiría desarrollar el sistema en cualquier lenguaje de programación, donde podamos enfocarnos en la velocidad de procesamiento por sobre lo disponible en el entorno web.

### 3.3. Tecnologías Relevantes al Trabajo

El lenguaje de programación que vamos a utilizar para realizar el trabajo es *Python*, debido a que es ampliamente utilizado en el campo de *Computer Vision* gracias a su flexibilidad, bibliotecas especializadas y facilidad de uso.

La principal herramienta que utilizaremos es *OpenCV*[1], una biblioteca de código abierto popular que proporciona una amplia gama de funciones y algoritmos para el procesamiento de imágenes y videos.

Para la construcción de la red neuronal convolucional haremos uso de *Keras*[2]. Esta es una biblioteca de código abierto con una interfaz de alto nivel para redes neuronales de la biblioteca *TensorFlow*[3], que usaremos para facilitarnos el entrenamiento y la evaluación de los modelos de aprendizaje profundo.

Para el armado del modelo haremos uso del *dataset* de imágenes *Open Images*[4], el cual proporciona una amplia colección de imágenes anotadas en diversas categorías. Este será una de las fuentes principales para entrenar nuestra red neuronal ya que nos permitirá trabajar con un conjunto diverso y representativo de imágenes.

Una tecnología que ya mencionamos y de la cual vamos a hacer uso es *WebAssembly*[5]. *WASM* nos va a permitir compilar el código fuente que realicemos en *Python* a un código que pueda correr directamente en un navegador web sin tener que resignar su rendimiento.

Durante el desarrollo inicial utilizaremos *Roboflow*[6], un producto que ofrece acceso a redes neuronales convolucionales de código abierto para clasificar imágenes y videos a través de llamados *HTTP*. Estas redes nos ayudarán en las pruebas de concepto y evaluación de opciones de desarrollo sin necesidad de crear nuestra propia red convolucional. Sin embargo, debido a la dependencia de un servicio externo y la latencia asociada, *Roboflow* no formará parte del trabajo final. De todas formas, los conjuntos de datos utilizados para entrenar las redes de *Roboflow* son públicos y se utilizarán como una de las fuentes para entrenar nuestra propia red.

## 4. Esquema de Trabajo

### 4.1. Metas Parciales

Considerando la naturaleza progresiva del problema, el trabajo se descompone en una serie de metas que se pueden agrupar de acuerdo a su dependencia y modularidad. Estos grupos de metas están planificados para que se puedan trabajar de manera independiente.

- **CV:** Partiendo de una red convolucional no propia, y tomándola como ya hecha, el trabajo propio de *Computer Vision* se descompone en las siguientes metas

CV1	Clasificar una imagen de acuerdo a si contiene o no contiene un cartel publicitario con una CNN de base.
CV2	Evaluar los frameworks de detección de objetos como SSD, YOLO y R-CNN. Detectar la porción de la imagen donde se contiene el cartel.
CV3	Descomponer un video en sus imágenes por segundo y procesar cada una de manera separada.
CV4	Evaluar las implementaciones de rastreo de objetos como CSRT, KCF y MOSSE. Detectar los objetos en algunas imágenes del video, y rastrear su movimiento, en vez de procesar cada imagen por separado.
CV5	Introducir modificaciones en las imágenes del video como difuminado y resaltado.
CV6	Optimizar la velocidad de procesamiento de video para poder procesar los videos en tiempo real y en dispositivos de bajo poder computacional.

- **CNN:** El modelo de red convolucional se va a desarrollar por separado para luego poder modificarlo para detectar cualquier objeto deseado en vez de restringirse a un solo dominio.

CNN1	Obtención de datos de carteles publicitarios.
CNN2	Evaluar las arquitecturas de redes convolucionales como VGGNet o MobileNet. Armar una red convolucional que detecta carteles publicitarios.
CNN3	Armar una red convolucional que detecta publicidades en general (por ejemplo, logos en ropa) en vez de solamente carteles.

- **WEB:** El sistema será desplegado en la web y accede a la cámara en tiempo real del cliente.

WEB1	Evaluar ventajas y desventajas de WASM vs Cliente/Servidor y confirmar la viabilidad de la elección.
WEB2	Armar un sitio web local que acceda a la cámara del usuario y utilice el modelo.
WEB3	Desplegar el servicio web a un servidor externo y evaluar la latencia y disponibilidad.

## 4.2. Cronograma Final

En base a las metas planteadas se arma un cronograma final de las distintas etapas que se van construyendo para llegar al trabajo deseado.

- **Prueba de Concepto y Viabilidad**

*Abarca las metas: CV1, CV2, CNN1*

La idea de esta etapa es confirmar que nuestra propuesta es viable y se va a poder hacer. Una vez confirmado que es posible hacer lo que buscamos, procederemos a la siguiente etapa del trabajo.

- **Arquitectura del Trabajo**

*Abarca las metas: WEB1, WEB2*

Buscamos evaluar las diferentes alternativas de despliegue y decidir la arquitectura final de nuestra solución. Seleccionaremos una, confirmando que es factible y que va a tener un rendimiento adecuado.

- **Trabajo final**

*Abarca las metas: WEB3, CNN3, CV5, CV6*

El paso final del trabajo es integrar las distintas partes que se fueron desarrollando y lograr hacer procesamiento de videos en tiempo real, desde un sitio web ya desplegado.

## Referencias

- [1] OpenCV: <https://opencv.org/>
- [2] Keras: <https://keras.io/>
- [3] TensorFlow: <https://www.tensorflow.org/>
- [4] Open Images Dataset: <https://storage.googleapis.com/openimages/web/index.html/>
- [5] WebAssembly: <https://webassembly.org/>
- [6] Roboflow: <https://roboflow.com/>

## Bibliografía

- Rosebrock, Adrian. (2017). *Deep learning for Computer Vision with Python*.
- Ansari, Shamshad. (2020). *Building Computer Vision Applications Using Artificial Neural Networks*, Apress.
- Moroney, Laurence. (2020). *AI and Machine Learning for Coders*, O'Reilly.

## Acta Acuerdo para el desarrollo de Trabajo Profesional

En Buenos Aires, al veinteavo día del mes de julio del año dos mil veintitrés se reúnen el profesor Ing. Martín Buchwald con los estudiantes de las carreras de Ingeniería en Informática y Licenciatura en Análisis de Sistemas, Srs. Federico del Mazo (Padrón N°: 100.029) y Casimiro Pastine (Padrón N°: 100.017) respectivamente, para tratar la elección y acuerdo del tema de Trabajo Profesional para el Ciclo Superior de la carrera pertinente.

Teniendo en cuenta la propuesta presentada por los alumnos más las observaciones y mejoras propuestas por el profesor, se ha acordado el plan de trabajo para el desarrollo e implementación del proyecto "*Sistema de detección en tiempo real de publicidad en la vía pública*" que figura en el documento adjunto.

El acuerdo consiste en las siguientes pautas:

- 1) Los alumnos Federico del Mazo y Casimiro Pastine realizarán todas las etapas para la construcción e implementación del sistema.
- 2) El trabajo a realizar será presentado para cumplir los requisitos de la materia Trabajo Profesional.
- 3) El profesor Ing. Martín Buchwald acepta la función de tutor para dicho trabajo.

  
36.684.576

Ing. Martín Buchwald

  
40230465

Sr. Federico del Mazo

  
39910135

Sr. Casimiro Pastine



Buenos Aires, 20 de Julio de 2023  
OBJETO: Presentación Propuesta Trabajo Profesional


Sr. Director del Departamento de Computación de  
la Facultad de Ingeniería de la Universidad de Buenos Aires.

Me dirijo a Usted a fin de elevar a su consideración el acta donde consta mi conformidad y la de los alumnos Federico del Mazo (Padrón N°: 100.029) y Casimiro Pastine (Padrón N°: 100.017), en la que se acuerda el tema de su Proyecto de Trabajo Profesional de las carreras de Ingeniería en Informática y Licenciatura en Análisis de Sistemas: *“Sistema de detección en tiempo real de publicidad en la vía pública”*

Se adjunta a la presente carta:

- 1) Propuesta de Trabajo Profesional;
- 2) Currículum Vitae Académico.

Quedando a su disposición.

  
36.684.576

Ing. Martín Buchwald

# Federico del Mazo

## Curriculum Vitae

fedede@fedede.dm

in FdelMazo ◊  fedede.dm ◊  FdelMazo

## Experiencia Laboral

---

### FREELANCER

- ♦ **DOS MONOS.**- Solución global de e-mail marketing

Python Javascript System Design

- ♦ **DEL FONDO EDITORIAL.**- Editorial de libros clásicos

L<sup>A</sup>T<sub>E</sub>X

**10PINES.**- Servicio de desarrollo de sistemas de software a medida.

- ♦ **LATTICE.**- The People Management Platform

Node.js GraphQL React

**RAICONET.**- Importación y exportación aérea y marítima.

Java Groovy Grails

### UNIVERSIDAD DE BUENOS AIRES, FACULTAD DE INGENIERÍA

- ♦ Algoritmos y Programación II - Curso Wachenchauzer

C Data Structures Memory Management

- ♦ Teoría de Algoritmos I - Curso Wachenchauzer

Computational Complexity Algorithm Heuristics

### Desarrollador de software

Noviembre 2021 – Presente

Julio 2020 – Enero 2021

Marzo 2020 – Enero 2022

Marzo 2018 – Marzo 2020

### Colaborador en materias

Agosto 2017 – Presente

Enero 2019 – Junio 2020

## Educación

---

### UNIVERSIDAD DE BUENOS AIRES, FACULTAD DE INGENIERÍA

Ingeniería en Informática (especialización en Sistemas Distribuidos)

2016 – Presente

### Colegio Ward

Bachiller Bilingüe en Economía y Administración

2009 – 2014

Julio 2023

**Alumno:** Federico del Mazo  
**Propuesta:** (10) Ingeniería en Informática  
**Versión:** 15

**Legajo:** 100029  
**Plan:** (1986) 1986

Actividad	Nota	Origen	Créditos
ANÁLISIS MATEMÁTICO A (CBC66)	5 (Aprobado)	Equivalencia	9.00
ALGEBRA A (CBC62)	9 (Aprobado)	Equivalencia	9.00
FÍSICA (CBC03)	7 (Aprobado)	Equivalencia	6.00
QUÍMICA (CBC05)	7 (Aprobado)	Equivalencia	6.00
INTRODUCCIÓN AL CONOCIMIENTO DE LA SOCIEDAD Y EL ESTADO (CBC24)	7 (Aprobado)	Equivalencia	4.00
INTRODUCCIÓN AL PENSAMIENTO CIENTÍFICO (CBC40)	9 (Aprobado)	Equivalencia	4.00
ANALISIS MATEMATICO II A (6103)	6 (Aprobado)	Examen	8.00
FISICA I A (6201)	4 (Aprobado)	Examen	8.00
ALGORITMOS Y PROGRAMACION I (7540)	8 (Aprobado)	Examen	6.00
ALGEBRA II A (6108)	9 (Aprobado)	Examen	8.00
FISICA II A (6203)	6 (Aprobado)	Examen	8.00
ALGORITMOS Y PROGRAMACION II (7541)	9 (Aprobado)	Examen	6.00
PROBABILIDAD Y ESTADISTICA B (6109)	7 (Aprobado)	Examen	6.00
ESTRUCTURA DEL COMPUTADOR (6670)	6 (Aprobado)	Examen	6.00
ALGORITMOS Y PROGRAMACION III (7507)	6 (Aprobado)	Examen	6.00
ANALISIS NUMERICO I (7512)	7 (Aprobado)	Examen	6.00
SISTEMAS OPERATIVOS (7508)	10 (Aprobado)	Examen	6.00
ORGANIZACION DE DATOS (7506)	10 (Aprobado)	Examen	6.00
TALLER DE PROGRAMACION I (7542)	7 (Aprobado)	Examen	4.00
MODELOS Y OPTIMIZACION I (7114)	6 (Aprobado)	Examen	6.00
TALLER DE PROGRAMACION II (7552)	10 (Aprobado)	Examen	4.00
ANALISIS DE LA INFORMACION (7509)	9 (Aprobado)	Examen	6.00
TECNICAS DE DISEÑO (7510)	6 (Aprobado)	Examen	6.00
BASE DE DATOS (7515)	7 (Aprobado)	Examen	6.00
INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS (7543)	6 (Aprobado)	Examen	6.00
ADM. Y CONTROL DE PROY. INFORMATICOS I (7544)	9 (Aprobado)	Examen	6.00
TECNICAS DE PROGRAMACION CONCURRENTES I (7559)	9 (Aprobado)	Examen	6.00
SISTEMAS DISTRIBUIDOS I (7574)	5 (Aprobado)	Examen	6.00
TALLER DE PROGRAMACION III (7561)	10 (Aprobado)	Examen	6.00
SIST.AUTOM.DE DIAG.Y DETEC.DE FALLAS I (7567)	8 (Aprobado)	Examen	6.00
MATEMATICA DISCRETA (6107)	4 (Aprobado)	Examen	6.00
QUIMICA (6301)	5 (Aprobado)	Examen	6.00
LABORATORIO (6602)	4 (Aprobado)	Examen	6.00
CRIPTOGRAFIA Y SEGURIDAD INFORMATICA (6669)	7 (Aprobado)	Examen	6.00
LENGUAJES FORMALES (7514)	9 (Aprobado)	Examen	6.00
SIMULACION (7526)	6 (Aprobado)	Examen	6.00
TEORIA DE ALGORITMOS I (7529)	10 (Aprobado)	Examen	6.00
TEORIA DE ALGORITMOS II (7530)	10 (Aprobado)	Examen	6.00
TEORIA DE LENGUAJE (7531)	10 (Aprobado)	Examen	4.00
ARQUITECTURA DEL SOFTWARE (7573)	9 (Aprobado)	Examen	4.00
ACTIVIDADES ACADEMICAS AFINES 6 (AAF6)		Aprob. por Resolución	6.00
IDIOMA INGLES (7801)	8 (Aprobado)	Examen	4.00

TOTAL DE CRÉDITOS APROBADOS	252.00
-----------------------------	--------

# Casimiro Pastine

[casimiropastine@gmail.com](mailto:casimiropastine@gmail.com)

## EXPERIENCIA

### **Pulpos:** *Desarrollador de Software*

Abril 2023 - Actualidad

- Desarrollo de funcionalidades dentro una aplicación de punto de venta para gestionar pequeños negocios.

Tecnologías: NodeJS, Trpc, NextJS, AWS, PostgreSQL

### **TheFork:** *Desarrollador de Software*

Diciembre 2019 - Abril 2023

- Desarrollo y mantenimiento de diversos microservicios utilizando NodeJS, PostgreSQL, GraphQL y JSON-RPC.
- Migré el sistema de atribución de tráfico de campañas de marketing y SEO a un stack tecnológico moderno, resolviendo diversos bugs y problemas que tenía el sistema viejo. Hoy en día el sistema ayuda a medir la performance de todas las campañas de marketing y SEO de la empresa.
- Participé en el desarrollo de la herramienta de AB Testing utilizada por todos los equipos de producto para desplegar nuevas funcionalidades con la confianza de tener un impacto positivo en el CVR.

Tecnologías: NodeJS, GraphQL, JSON-RPC, RabbitMQ, PostgreSQL, React

### **Globant:** *Desarrollador de Software*

Julio 2018 - Agosto 2019

- Desarrollo y mantenimiento de una plataforma de aprovisionamiento de infraestructura para J.P.Morgan.

Tecnologías: Python, Django, PyTest, Git.

## EDUCACIÓN

### **Facultad de Ingeniería, UBA:** *Licenciatura en Análisis de Sistemas*

2015 - Now

### **Instituto Pedro Poveda:** *Colegio Secundario*

2009 - 2014

## TRABAJO VOLUNTARIO

### **Facultad de Ingeniería, UBA**

Aug 2017 - Dic 2019

- Colaborador en Algoritmos y Programación I y II, curso Wachenchauser.

**Alumno:** Casimiro Pastine

**Legajo:** 100017

**Propuesta:** (9) Licenciatura en Análisis de Sistemas

**Plan:** (86V14) 86V14

Actividad	Nota	Origen	Créditos
FÍSICA (CBC03)	6 (Aprobado)	Equivalencia	6
INTRODUCCIÓN AL CONOCIMIENTO DE LA SOCIEDAD Y EL ESTADO (CBC24)	7 (Aprobado)	Equivalencia	4
INTRODUCCIÓN AL PENSAMIENTO CIENTÍFICO (CBC40)	7 (Aprobado)	Equivalencia	4
QUÍMICA (CBC05)	5 (Aprobado)	Equivalencia	6
ALGEBRA A (CBC62)	5 (Aprobado)	Equivalencia	9
ANÁLISIS MATEMÁTICO A (CBC66)	9 (Aprobado)	Equivalencia	9
ALGORITMOS Y PROGRAMACION I (9514)	9 (Aprobado)	Equivalencia	6
MATEMÁTICA DISCRETA (8111)	8 (Aprobado)	Equivalencia	6
ORGANIZACIÓN DEL COMPUTADOR (9557)	7 (Aprobado)	Examen	8
ALGORITMOS Y PROGRAMACIÓN II (9515)	9 (Aprobado)	Equivalencia	6
ANÁLISIS MATEMÁTICO II (8101)	10 (Aprobado)	Equivalencia	8
ÁLGEBRA II (8102)	8 (Aprobado)	Equivalencia	8
ECONOMÍA DE LAS ORGANIZACIONES (9141)	9 (Aprobado)	Examen	4
ESTRUCTURAS Y PROCESOS ORGANIZACIONALES (9139)	8 (Aprobado)	Examen	6
ORGANIZACIÓN DE DATOS (9558)	9 (Aprobado)	Examen	6
PROBABILIDAD Y ESTADÍSTICA B (8104)	9 (Aprobado)	Examen	6
ALGORITMOS Y PROGRAMACIÓN III (9502)	7 (Aprobado)	Equivalencia	6
TALLER DE PROGRAMACIÓN (9508)	7 (Aprobado)	Examen	4
MÉTODOS Y MODELOS EN LA INGENIERÍA DE SOFTWARE I (9520)	6 (Aprobado)	Examen	6
SISTEMAS OPERATIVOS (9503)	8 (Aprobado)	Examen	6
REDES Y APLICACIONES DISTRIBUIDAS (9560)	7 (Aprobado)	Examen	6
MÉTODOS Y MODELOS EN LA INGENIERIA DE SOFTWARE II (9521)	7 (Aprobado)	Examen	6
ADMINISTRACIÓN DE LAS ORGANIZACIONES (9142)	9 (Aprobado)	Examen	4
MODELOS Y OPTIMIZACIÓN I (9104)	8 (Aprobado)	Examen	6
BASE DE DATOS (9505)	9 (Aprobado)	Examen	6
CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA (8636)	6 (Aprobado)	Examen	6
TEORÍA DE LENGUAJES DE PROGRAMACIÓN (9507)	9 (Aprobado)	Examen	4
INTELIGENCIA ARTIFICIAL (9525)	9 (Aprobado)	Examen	6
DESARROLLO CON NUEVAS TECNOLOGÍAS (9568)	8 (Aprobado)	Regularidad	4

Total de créditos aprobados	172
-----------------------------	-----