

Escuela de Código para PILARES

Tabla de especificaciones e Instrumentos de evaluación

Programación (MP)



Escuela de Código para PILARES Tabla de especificaciones e Instrumentos de evaluación - MP por Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México se distribuye bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

Créditos¹

Coordinación de módulos

Adrián Durán Chavesti, Helena Gómez Adorno, Ivan Vladimir Meza Ruiz, Nora Isabel Pérez Quesadas, María del Pilar Ángeles, Víctor Manuel Lomas Barrie, Zian Fanti Gutierrez

Coordinación de la Transversalización de la Perspectiva de Género

Yuliana Ivette López Rodríguez

Revisiones

Karen Itzel Bruno Sainos, Citlalli Sánchez Mendoza, Carmen Daniela Garrido Juvencio

Supervisión PILARES

Jesús Alanis Manriquez, René Alejandro Rivas Robles y María del Rocío Estrada Monroy

Supervisión IIMAS

Alejandra Sarahí Monroy Velázquez, Andrea García Ruiz, Carla Irena Blenda Palacios, Elisa Mariana Valdés Armada, Héctor Alfonso Islas García, Héctor Benítez Pérez, Karen Alexa Alva Aguirre, Karina Flores García y Luz Elena Rueda Rojas

Financiamiento:

Diseño de un programa de estudios para la capacitación en programación y habilidades en tecnologías de información y comunicación para la escuela de código dentro de PILARES de la Ciudad de México (SECTEI/284/2019).

¹ En orden alfabético.

Agradecimientos

Agradecemos el tiempo y la retroalimentación hecha a los materiales a:

- Ante Salcedo González, ITAM - Instituto Tecnológico Autónomo de México
- Blanca Esther Carvajal-Gámez, ESCOM - Escuela Superior de Cómputo - IPN
- Dagoberto Pulido Arias, IPN - Instituto Politécnico Nacional
- Eréndira Itzel García Islas, UNAM - Facultad de Ciencias
- Marco Antonio Moreno Ibarra, CIC - Centro de Investigación en Computación del IPN
- Ricardo Marcelín Jiménez, UAM-I Universidad Autónoma Metropolitana – Iztapalapa
- Salvador Elias Venegas Andraca, ITESM - Instituto Tecnológico y de Estudios Superiores de Monterrey

También agradecemos el apoyo y seguimiento al personal de SECTEI, en particular de:

- José Bernardo Rosas Fernandez
- Federico Antonio Hernández Loranca
- Rogelio Artemio Morales Martínez
- Adrián Eleazar Contreras Martínez
- Benigno Antonio González Núñez

Índice	
Créditos	2
Agradecimientos	3
Índice	3
Introducción	5
Programación	6
Presentación	6
Competencias y productos	6
Ejes a evaluar	7
Tabla de especificación: nivel y aspecto/instrumento	9
Tabla de cálculo de nivel de competencia por taller	10
Tabla de cálculo de competencia por módulo	10
Instrumentos de evaluación: comunes entre los talleres	11
Rúbrica: Reporte del proyecto	11
Lista de verificación: Actitudes	13
Lista de verificación: Perspectiva de género	14
Instrumentos de evaluación: específicos a los talleres	14
Taller 1: Aprende a programar jugando Ajedrez	14
Cuestionario: Conocimiento de conceptos y herramientas	14
Respuestas	19
Lista de cotejo: Funcionalidad de proyecto	20
Taller 2: Construyendo una aplicación web con Python	21
Cuestionario: Conocimiento de conceptos y herramientas	21
Respuestas	25
Lista de cotejo: Funcionalidad de proyecto	25
Taller 3: Introducción a ciencia de datos con Python	27
Cuestionario: Conocimiento de conceptos y herramientas	27
Lista de cotejo: Funcionalidad de proyecto	33

Introducción

El objetivo de este documento es ofrecer una guía de cómo realizar la evaluación de los talleres y módulos. El presente documento está conformado por las tablas de especificación que ayudan a determinar el nivel de competencia de la participante por módulo y los instrumentos de evaluación a ser utilizados para evaluar el desempeño de la participante.

Por cada módulo se incluye una presentación del módulo, los ejes a evaluar, la tabla de especificación a nivel taller y módulo, los instrumentos de evaluación comunes en el taller y los instrumentos de evaluación específicos al taller.

Los instrumentos comunes están compuestos por:

- Rúbrica de presentación del proyecto.
- Listas de verificación de actitudes y perspectiva de género.

Los instrumentos específicos al taller son:

- El cuestionario de conocimientos.
- La lista de cotejo de funcionalidad del proyecto.

Programación

Presentación

El módulo de *Programación* prepara a las participantes para que sean capaces de implementar aplicaciones gráficas, web o manejar datos en el lenguaje de programación python bajo supervisión. Para tomar el módulo es necesario que la participante haya sido competente en el *Módulo cero: conociendo el ambiente de computación*. Este módulo se divide en tres talleres durante los cuales son guiadas por una tallerista y que culminan con un producto (proyecto) en la temática del taller y módulo. Los talleres permiten a la participante adquirir el conocimiento técnico y ponerlo en práctica. Además durante los talleres se promueve la demostración de las actitudes pertinentes para el desarrollo laboral, la reflexión alrededor de temáticas de perspectiva de género y la práctica de la autonomía de la participante. En el taller uno la participante se enfoca el aprendizaje del lenguaje implementando un juego de ajedrez, en taller dos se enfoca a la creación de una aplicación web para el manejo de una biblioteca y finalmente en el taller tres presenta una introducción a la ciencia de datos.

Este documento tiene el objetivo de presentar la tabla de especificación y los instrumentos de evaluación del módulo de *Programación*. La evaluación se hará en el esquema de competencias por cada taller, al final se acumulará estas tres evaluaciones para conformar la evaluación global del módulo. Para que una participante se haga acreedora a una evaluación deberá haber hecho todas las actividades correspondiente al taller, de hecho, la participante no podrá avanzar a la siguiente actividad sino ha concluido la actividad anterior.

La evaluación consistirá en la determinación del nivel de competencia basado en aspectos cuantificables y una retroalimentación a aspectos de crecimiento personal que benefician su desempeño laboral. Los componentes de la evaluación y los niveles de esta se especifican en la sección *Tabla de especificación*; los instrumentos propuestos se listan por taller en la sección *Instrumentos de Evaluación*. Antes de proceder con estos elementos se listan las competencias y los productos esperados en el módulo.

Competencias y productos

La competencia de este módulo es:

Resolución de problemas a través de la implementación de proyectos de programación en el lenguaje Python en tres diferentes facetas: aplicaciones de escritorio, aplicaciones web y análisis de datos.

La siguiente tabla resume las competencias y productos (proyectos) por taller.

Taller uno: Aprende a programar jugando Ajedrez	
Competencia	Programar un juego de ajedrez en Python que haga uso de diferentes tipos de variables, arreglos, y listas, con controles de secuencia como son el if, else, while y operaciones con número y caracteres.
Producto	Juego de ajedrez usando la librería pygame.
Taller dos: Construyendo una aplicación web con Python	
Competencia	A través de una plataforma enfocada a la creación de aplicaciones web, la participante desarrollará un sistema para la administración de una biblioteca utilizando el lenguaje de programación Python.
Producto	Webapp para administración de préstamos de libros en biblioteca usando la librería flask.
Taller tres: Introducción a ciencia de datos con Python	
Competencia	Procesar, manipular, analizar y visualizar colecciones de datos usando el lenguaje de programación python y el ambiente de desarrollo de Notebooks Jupyter.
Producto	Análisis de datos abiertos usando la interfaz tipo Notebook Jupyter y librería Pandas.

Ejes a evaluar

La evaluación de la participante se realizará en tres ejes: el técnico *¿qué sabe?*, el de proyecto *¿cómo lo aplica?* y de crecimiento profesional *¿ha demostrado comportamiento conducente al desempeño profesional?* Tanto el primer como el segundo eje contribuyen a determinar el nivel de competencia de la participante. El primer eje se enfoca a cuantificar el dominio de los conceptos relacionados a programación, mientras que el segundo se enfoca a cuantificar cómo usa ese conocimiento para hacer un juego, un aplicación web y análisis de datos. El tercer eje corresponde a aspectos no cuantificables pero que apoyan a un desarrollo profesional de la participante por taller.

El eje técnico está compuesto por los siguientes aspectos a evaluar:

- Conocimiento de conceptos y herramientas: En este aspecto el conjunto de conocimiento adquirido durante un taller.

El eje del proyecto está compuesto por los siguientes aspectos a evaluar:

- Funcionalidad del proyecto: Este aspecto se enfoca a evaluar el proyecto/producto alcanzado al final de su taller. La evaluación se divide en dos etapas, primero que tenga los componentes solicitados y segundo que las relaciones entre estos elementos estén establecidas dada la especificación. La composición de estas etapas nos permitirá evaluar la funcionalidad del proyecto.
- Reporte del proyecto: Este aspecto se enfoca a que la participante tenga la capacidad de abstraer, narrar y expresar el proceso de la realización de su proyecto. En el taller uno se solicitará una presentación oral una a una, participante-tallerista; en el taller dos será de nuevo una presentación oral pero una a muchas, participante-un público; y en el taller tres además de la presentación una a muchas, elaborará un reporte corto escrito de su proyecto.

Finalmente el eje de crecimiento profesional está compuesto por:

- Demostración de actitudes: Este aspecto se enfoca verificar la demostración de las siguientes actitudes:
 - Paciencia: La participante sorteará contratiempos y dificultades para la realización de las actividades.
 - Perspicacia: La participante comprenderá, deducirá y razonará sobre los conceptos que le permitan realizar las actividades.
 - Curiosidad: La participante pregunta sobre temáticas del taller o relacionadas.
 - Disposición: La participante demuestra disponibilidad para realizar las actividades.
 - Constancia: La participante realiza las actividades de forma secuencial sin interrupciones.
 - Persistencia: La participante ante dificultades sigue intentando realizar las actividades.
- Comprensión de temas de perspectiva de género: Dado que este es un eje transversal a los módulos, la evaluación se enfoca a la apertura y entendimiento que adquiere la participante a través de diversas actividades en la temática de perspectiva de género.
- Autonomía: Este es otro eje transversal de los módulos, en particular se busca que la participante crezca en su autonomía de estudio. Comenzará en el taller uno, etapa durante la cual se le guiará en cómo encontrar las respuestas para las problemáticas planteadas; para el taller dos, se le indica el dónde encontrar esas respuestas y para el taller tres se deja a la participante encontrar las fuentes que le permitan resolver su problemática. Aunque este aspecto aboga hacia la autonomía de la participante, el

objetivo no es alcanzar una autonomía total, ya que el perfil de egreso nos indica que la participante se desempeñará en lo laboral bajo supervisión.

Tabla de especificación: nivel y aspecto/instrumento

Los requisitos para que un taller sea evaluado son los siguientes:

- Haber realizado todas las actividades del taller.

La siguiente tabla especifica los niveles de competencia por aspecto a evaluar y determina el instrumento a utilizar.

	Nivel competencia			
Aspecto/ Instrumento	No evaluable	Incipiente	En desarrollo	Consolidado
Conocimiento partes/ Cuestionario	- Menos de ocho reactivos correctos	- Entre ocho y doce reactivos correctos (9-12)	- Entre trece y diecisiete reactivos correctos (13-17)	- Más de dieciocho reactivos correctos (18-20)
Funcionamiento del proyecto/ Lista de cotejo	- No hay proyecto a evaluar	- Los elementos del proyecto funcionan de forma independiente.	Los elementos del proyecto están integrados pero hay algunas integraciones que fallan.	-Los elementos del proyecto están integrados y no presentan ninguna falla.
Reporte del proyecto/ Presentación	- No presenta	- Sólo responde a ¿qué hace el proyecto?	- Adiciona: * Objetivo. * Motivación.	- Adiciona: * Casos de uso. * Arquitectura. * Impacto de solución.
Actitudes/ Lista de verificación	La recomendación será sobre aquellas actitudes que no hayan sido demostradas durante un taller. La retroalimentación será positiva y sólo se explicará como una alternativa para mejorar su experiencia laboral.			
Perspectiva de género/ Lista de verificación	En caso de poca o nula participación se dará una retroalimentación que se ha notado la indiferencia al tema.			
Independencia/ No aplica	Los talleres abordan la autonomía de la siguiente forma: - Taller 1: Seguía la resolución de problemas y respuesta a preguntas. - Taller 2: Se deja a la participante la resolución de problemas y respuestas a preguntas			

	<p>con las fuentes proporcionadas.</p> <p>- Taller 3: La participante tiene que buscar cómo resolver los problemas y las fuentes para las respuestas a preguntas.</p> <p>La retroalimentación al final de cada taller consiste en hacer consciente a la participante del nivel de autonomía que tuvo en el taller y el que tendrá que enfrentarse en el siguiente.</p>
--	--

Tabla de cálculo de nivel de competencia por taller

La siguiente tabla establece el cálculo de la competencia por taller, cabe recordar que el nivel de competencia sólo está establecido por cuatro aspectos de los ejes técnicos y de proyecto.

No de no evaluable	Número de incipientes	Número de en desarrollo	Número en consolidado	Final
0	3	0	0	Incipiente
0	2	1	0	Incipiente
0	2	0	1	Incipiente
0	0	3	0	En desarrollo
0	1	2	0	En desarrollo
0	0	2	1	En desarrollo
0	1	1	1	En desarrollo
0	1	0	2	En desarrollo
0	0	0	3	Consolidado
0	0	1	2	Consolidado
>0	*	*	*	No evaluable

Tabla de cálculo de competencia por módulo

La siguiente tabla establece el cálculo de la competencia del módulo, en este cálculo se toman en cuenta las tres competencias calculadas por taller.

No de no	Número de	Número de en	Número en	Final
----------	-----------	--------------	-----------	-------

evaluable	incipientes	desarrollo	consolidado	
0	3	0	0	Incipiente
0	2	1	0	Incipiente
0	2	0	1	Incipiente
0	1	2	0	En desarrollo
0	1	1	1	En desarrollo
0	0	3	0	En desarrollo
0	0	2	1	En desarrollo
0	1	0	2	En desarrollo
0	0	1	2	Consolidado
0	0	0	3	Consolidado
>0	*	*	*	No evaluable

Instrumentos de evaluación: comunes entre los talleres

Rúbrica: Reporte del proyecto

La participante presentará su proyecto a la tallerista durante su presentación:

1. Taller uno: presentación oral entre participante y tallerista.
2. Taller dos: presentación oral entre participante y grupo.
3. Taller tres: presentación escrita.

Los específicos de la presentación serán acordados en conjunto con la tallerista, pero en cualquier modalidad de la presentación se deberán responder a las siguientes preguntas detonadoras:

- ¿De qué se trata el proyecto? (Tema)
- ¿Cuál es el objetivo del proyecto? (Objetivo)
- ¿Cuál es la razón de hacer dicho proyecto? (Motivación)
- ¿Cuales son los elementos del proyecto y cómo se organizan éstos? (Arquitectura)
- ¿Cómo se usa el juego? (Demostración)

A continuación se define la rúbrica de evaluación:

	No evaluable	Incipiente	En desarrollo	Consolidado
Tema	No presenta	Repite temática del taller y proyecto. Ej. <i>Un juego de ajedrez.</i>	Extiende la temática del taller y proyecto. Ej. <i>Un juego de ajedrez interactivo a través de la interfaz gráfica.</i>	Parafrasea temática del taller y proyecto. <i>ChampionChess: juego de ajedrez para principiantes.</i>
Objetivo	No presenta	Repite temática del taller y proyecto. Ej. <i>Hacer un juego de ajedrez.</i>	Parafrasea la temática del taller y proyecto. Ej. <i>Hacer una interfaz gráfica para jugar ajedrez.</i>	Define el propósito de su proyecto usando sus propias palabras. Ej. <i>Permitir a jugadores de ajedrez jugar en la computadora.</i>
Motivación	No presenta	Ancla razón alrededor de la temática del taller y proyecto. Ej. <i>Poder jugar ajedrez.</i>	Establece una razón anclada en el propósito general del proyecto. Ej. <i>Jugar ajedrez en la computadora usando la interfaz gráfica.</i>	Establece una razón anclada en el propósito y beneficio del proyecto. Ej. <i>Permitir a los jugadores jugar ajedrez en la computadora de una forma intuitiva.</i>
Arquitectura	No presenta	Enumera las partes de su proyecto. Ej. <i>El proyecto tiene</i>	Enuncia las partes y establece relaciones entre éstas en su	Enuncia las partes, las agrupa en módulos y

		<i>piezas, reglas, ...</i>	<i>proyecto. Ej. Ej juego permite varias modalidades de juego.</i>	<i>establece relaciones entre partes y módulos de sus proyecto. Ej. El juego tiene tres módulos principales, ...</i>
Demostración	No presenta	Describe elementos de su proyecto. Ej. <i>La ventana principal tiene un menú, ...</i>	Presenta su proyecto de forma directa. Ej. <i>Presenta su proyecto ejecutandolo en vivo y demostrándolo.</i>	Presenta y narra partes de su proyecto en forma indirecta. Ej. <i>Presenta su proyecto describiendo el flujo de este y usando ayudas visuales como capturas de pantalla.</i>

Lista de verificación: Actitudes

La tallerista llevará un registro de las ocasiones que la participante demuestre alguna de las siguientes actitudes:

Paciencia, perspicacia, curiosidad, constancia y persistencia.

Para este registro recomendamos el formato para la siguiente tabla:

Actitud	Actividad/Taller	Fecha	Observaciones

Al final del taller, la tallerista identificará aquellas actitudes no demostradas o con pocas ocurrencias y podrá elaborar una retroalimentación personalizada recomendando una estrategia para incorporar la actitud en el ámbito profesional.

Lista de verificación: Perspectiva de género

La tallerista llevará un registro de las ocasiones en que la participante sea parte de las actividades referentes a perspectiva de género. Para este registro recomendamos una tabla, donde pueda anotar la participación por participante.

Al final del taller, la tallerista identificará si hubo poca o nula participación, dará una retroalimentación que se ha notado la indiferencia al tema.

Instrumentos de evaluación: específicos a los talleres

Taller 1: Aprende a programar jugando Ajedrez

Cuestionario: Conocimiento de conceptos y herramientas

Consistirá en 20 reactivos de opción múltiple en los siguientes temas:

1. Lenguajes de programación.
2. Sintaxis.
3. Variables.
4. Control de flujo condicional.
5. Control de flujo bucles.
6. Funciones.
7. Objetos.
8. Módulos.
9. PyGames.

Reactivos

1. Son lenguajes de programación:
 - a. Python, C++, Java y Android
 - b. Python, C++, HTML y PHP
 - c. Python, pycharm, Javascript y PHP
 - d. C++, Java, PHP y Javascript
2. ¿Cuál es el flujo de conversión de un código de alto nivel que sigue un intérprete?
 - a. Código fuente → Compilador → Código máquina → Salida
 - b. Código fuente → Intérprete → Salida
 - c. Código fuente → Compilador → Intérprete → Salida

d. Código máquina →Intérprete → Salida

3. El siguiente código qué genera:

```
if 10 > 2:
```

```
    print("¡Diez es mayor a 2!")
```

- a. El texto "¡Diez es mayor a 2!"
- b. Un error.
- c. No genera nada.
- d. Genera el código máquina pero no lo ejecuta.

4. El siguiente código qué imprime:

```
"""
```

```
    Escuela de Código
```

```
    Pilares
```

```
    CDMX
```

```
"""
```

```
print("¡Hola, que tal!")
```

```
#print("¿Cómo estás?")
```

- a. ¡Hola, que tal!
- b. ¿Cómo estás?
- c. Escuela de Código.
- d. Nada.

5. El siguiente código qué imprime:

```
x = 5
```

```
y = 6
```

```
print(5 + x)
```

- a. 5 + x
- b. 11
- c. 5
- d. 10

6. Si a = [1, 2, 3, 4, 5], cuánto debe valer i para que se cumpla a[i]=4

- a. 3
- b. 4
- c. 5
- d. len(a)-1

7. El siguiente código qué imprime:

```
a = "¡Hola, que tal!"
```

```
print(len(a)-1)
```

- a. len(a)-1
- b. a
- c. 14
- d. 15

8. El siguiente código qué imprime:

```
tupla = ("abc", 2, True, 40, "manzana")  
a = tupla[1]  
print(tupla[a])
```

- a. 2
- b. True
- c. "abc", 2, True, 40, "manzana"
- d. 40

9. El siguiente código qué imprime:

```
diccionario = {  
  
    "nombre": "Adriana",  
    "registrada": True,  
    "fecha": 1964,  
    "hermanos": ["Rosa", "Juan", "Enrique"]  
}  
a = 1  
print(diccionario["hermanos"][a])
```

- a. Adriana
- b. "Rosa", "Juan", "Enrique"
- c. Juan
- d. Rosa

10. El siguiente código qué imprime:

```
diccionario = {  
    "nombre": "Adriana",  
    "registrada": True,  
    "fecha": 1964,  
    "hermanos": ["Rosa", "Juan", "Enrique"]  
}  
a = False  
diccionario.update({"registrada": a})  
print(diccionario["registrada"])
```

- a. a
- b. False
- c. True
- d. registrada

11. El siguiente código qué imprime:

```
a = 35
```



```
nueve = 9
print(a > 9)
a. False
b. True
c. 9
d. 35
```

12. El código siguiente ¿qué color imprime?

```
a = 5
if a > 5:

    print("rojo")
elif a < 5:

    print("verde")
else:

    print ("azul")
```

- a. rojo.
- b. verde.
- c. azul.
- d. rojoazul.

13. El código siguiente qué números imprime:

```
i = 1
while i < 6:

    print(i)
    i = i +1
```

- a. 1, 2, 3, 4, 5
- b. 0,1, 2, 3, 4, 5
- c. 0,1, 2, 3, 4, 5
- d. 1, 2, 3, 4

14. El código siguiente qué números imprime:

```
i = 1
while i < 6:

    print(i)
    if (i == 3):

        break
    i += 1
```

- a. 1, 2, 3
- b. 1, 2, 4, 5, 6

- c. 0, 1, 2, 3
- d. 1, 2, 2.9

15. El siguiente código qué frutas imprime:

```
frutas = ["plátano", "pera", "piña"]  
for x in frutas:
```

```
    print(x)
```

- a. platano, pera, pina
- b. x
- c. plátano, piña, pera
- d. plátano, pera, piña

16. El siguiente código qué imprime:

```
def suma(a, b):
```

```
    print(a + b)
```

```
a = 4
```

```
b = 5
```

```
suma(a, 6)
```

- a. a + b
- b. 10
- c. 9
- d. a + 6

17. El siguiente código qué imprime:

```
def imprime_frutas(comida):
```

```
    for x in comida:
```

```
        print(x)
```

```
frutas = ["uva", "melón", "ciruela"]
```

```
imprime_fruta(frutas)
```

- a. uva, melón, ciruela
- b. comida
- c. Error
- d. frutas

18. El siguiente código qué imprime:

```
class Persona:
```

```
    def __init__(self, nombre, edad):
```

```
        self.nombre = nombre
```

```
        self.edad = edad
```

```
nombre = "Claudia"
```

```
edad = 34
```

```
p1 = Persona("Fabiola", 22)
```

```
print(p1.nombre, ",", p1.edad )
```

a. Claudia , 34

b. Fabiola , 22

c. Fabiola , 34

d. Claudia , 22

19. ¿Para qué sirve un módulo en Python?

a. Para organizar mejor el código y tener una biblioteca de funciones.

b. Para programar mi código principal ahí.

c. Para programar toda la funcionalidad de mi programa ahí.

d. Para guardar variables.

20. ¿Qué componentes pertenecen a PyGames?

a. Ventanas, menús, botones y *splits*.

b. Monitor, ventanas, menús, botones.

c. Ventanas, menús, pixeles y *splits*.

d. Movimiento, teclado, mouse y sonido.

Respuestas

1. d

2. b

3. a

4. a

5. d

6. a

7. c

8. b

9. c

10. b

11. b

12. c

13. a

14. a

15. d

16. b

17. a

18. b

19. b

20. ??

Lista de cotejo: Funcionalidad de proyecto

Para evaluar la funcionalidad del proyecto la tallerista cotejará contra la siguiente lista:

Funcionalidad	SI	NO	Comentarios
Componentes de la interfaz			
Existen las imágenes de las piezas.			
Existe una función para mostrar el tablero.			
Existe una función para limpiar el tablero.			
Existe una clase pieza.			
Existen 32 objetos pieza.			
Atributos de la interfaz			
Cada objeto tiene propiedades de bando, tipo de pieza, imagen, posición inicial y posición actual.			
Cada objeto tiene un método mostrar, mover y eliminar una pieza.			
Métodos para controlar los componentes			
Existe un método para validar el movimiento de la reina y de la torre.			
Existe un método para capturar a la reina y eliminarla del tablero.			
Existe un método para indicar que hay jaque o jaque mate.			
Existe un menú con la siguientes opciones: Inicio -> reiniciar partida/salir Configuración -> Color casillas/Color piezas			

Ayuda -> Muestra una ventana con ayuda y créditos			
---	--	--	--

Taller 2: Construyendo una aplicación web con Python

Cuestionario: Conocimiento de conceptos y herramientas

Consistirá en 20 reactivos de opción múltiple en los siguientes temas:

1. Servidor web.
2. HTML.
3. HTTP.
4. Aplicaciones web.
5. Flask.
6. MongoDB.

Reactivos

1. En una página web estática:
 - a. El contenido sólo se modifica editando la página desde el servidor.
 - b. Las imágenes no se mueven.
 - c. Cada interacción con la página genera asincrónamente nuevo contenido.
 - d. No se pueden usar formularios.
2. Un servidor web es:
 - a. Un hardware para mostrar solo imágenes.
 - b. Un programa que genera archivos que pueden ser leídos por un navegador.
 - c. Una aplicación para consultar internet.
 - d. Un software para enviar correo electrónico.
3. Con este software no puedo implementar un servidor web:
 - a. Flask.
 - b. Apache.
 - c. HTML.
 - d. NGINX.
4. ¿Cuál es el lenguaje que interpreta un navegador para mostrar contenido (texto, imágenes, botones, ventanas, etc)?
 - a. Javascript
 - b. HTML.
 - c. JSON.
 - d. Python.
5. HTTP es un protocolo para:
 - a. Enlazar llamadas telefónicas digitales.
 - b. Comunicar a dos computadoras para que ejecute Python.

- c. Comunicar a dos computadoras el correo electrónico.
 - d. Comunicar a dos computadoras el contenido de una página web.
6. ¿Cuál de los siguientes incisos son métodos HTTP?
- a. GET, POST, ADD, UPDATE.
 - b. GET, POST, PUT, DELETE.
 - c. READ, ADD, UPDATE, ERASE.
 - d. GET, POST, PUT, ERASE.
7. ¿Qué es una dirección IP?
- a. Un número para identificar una aplicación web.
 - b. Un número para identificar una computadora.
 - c. Un número para identificar un programa de cómputo.
 - d. Es una URL para traducir a una dirección.
8. De los siguientes, ¿cuál no es un framework para aplicaciones web?
- a. Flask.
 - b. Django.
 - c. Cakephp.
 - d. Javascript.
9. ¿HTML es un lenguaje de programación?
- a. Cierto.
 - b. Solo la parte donde gestiona el comportamiento de las ventanas.
 - c. Falso.
 - d. No, es un programa para hacer textos.
10. ¿Qué etiqueta no forma parte de HTML?
- a. <head>
 - b. </title>
 - c. <script>
 - d. /body>
11. Un *template* en flask sirve para:
- a. Gestionar los recursos de flask.
 - b. Generar contenido web sin tener que escribir todas las variantes.
 - c. Identificar la ruta de una función.
 - d. Hacer páginas web fijas.

12. ¿Cuál es la URL correcta del siguiente recurso:

Protocolo: HTTPS

Puerto: 529

Dirección de la computadora: 216.58.193.4

Ruta del recurso: /busqueda/animales

Cadena de búsqueda: ?nombre=codorniz&aves=True

- a. <http://216.58.193.4:529/busqueda/animales?nombre=codorniz&aves=True>
 - b. <https://216.58.193.4:529/busqueda/animales?nombre=codorniz&aves=True>
 - c. <https://216.58.193.4/busqueda/animales?nombre=codorniz&aves=True>
 - d. <http://216.58.193.4/busqueda/animales?nombre=codorniz&aves=True>
13. ¿Cuál es la URL correcta para que el siguiente segmento de código en flask devuelva el título del libro con id = 4? Suponga que el recurso está en <http://www.biblio.org/>

```
@app.route('/libros/titulo_por_id/<int:libro_id>')
```

```
def titulo_libro(libro_id):
```

```
...
```

```
...
```

```
return 'Título del libro: %s' % titulo
```

- a. http://www.biblio.org/libros/titulo_por_id/?libro_id=4
 - b. http://www.biblio.org/libros/titulo_por_id/4
 - c. <http://www.biblio.org/4>
 - d. http://www.biblio.org/libros/titulo_por_id=4
14. ¿Qué es “Bootstrap”?
- a. Es una biblioteca de hojas de estilos CSS.
 - b. Un *framework* simplista para generación de páginas web.
 - c. Es un navegador.
 - d. Es un *framework* que se ejecuta en el *backend*.
15. ¿A qué se refiere el acrónimo CRUD?
- a. Son las funciones básicas en una tienda de mercadeo en línea.
 - b. Son las funciones básicas en un sistema para crear, leer, actualizar y borrar datos.
 - c. Son las funciones básicas en un sistema para crear, relacionar, ubicar y determinar.
 - d. Es lo que hace una página escrita en HTML.
16. Para el código que se muestra, indique la URL que se conforma al llamar a la función `print(url_for('login', next='/'))`. Suponga que el recurso está en <http://www.biblio.org/>

```
from flask import Flask, url_for
from markupsafe import escape

app = Flask(__name__)

@app.route('/')
def index():
    return 'index'

@app.route('/login')
def login():
    return 'login'

@app.route('/user/<username>')
def profile(username):
    return '{}\n's profile'.format(escape(username))
```

- a. <http://www.biblio.org/login>
 - b. <http://www.biblio.org/login?next=/>
 - c. <http://www.biblio.org/?next=/>
 - d. <http://www.biblio.org/>
17. ¿Cuál de los siguientes incisos es un motor de bases de datos NoSQL?
- a. MariaDB.
 - b. MongoDB.
 - c. MySQL.
 - d. Oracle.
18. ¿Cuál inciso representa mejor el nivel de los componentes en una base de datos NoSQL?
- a. Base de datos -> Tabla -> Documento -> Clave/campo
 - b. Base de datos -> Colección -> Documento -> Columna
 - c. Base de datos -> Colección -> Documento -> Clave/campo
 - d. Base de datos -> Tabla -> Registro -> Columna
19. Dado el código de Python que se muestra abajo, ¿cuál es la instrucción correcta para extraer solo los campos de "name" y "address"?

```
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]

mylist = [
    { "_id": 1, "name": "John", "address": "Highway 37"},
    { "_id": 2, "name": "Peter", "address": "Lowstreet 27"},
    { "_id": 3, "name": "Amy", "address": "Apple st 652"},
    { "_id": 4, "name": "Hannah", "address": "Mountain 21"},
    { "_id": 5, "name": "Michael", "address": "Valley 345"},
    { "_id": 6, "name": "Sandy", "address": "Ocean blvd 2"},
    { "_id": 7, "name": "Betty", "address": "Green Grass 1"},
    { "_id": 8, "name": "Richard", "address": "Sky st 331"},
    { "_id": 9, "name": "Susan", "address": "One way 98"},
    { "_id": 10, "name": "Vicky", "address": "Yellow Garden 2"},
    { "_id": 11, "name": "Ben", "address": "Park Lane 38"},
    { "_id": 12, "name": "William", "address": "Central st 954"},
    { "_id": 13, "name": "Chuck", "address": "Main Road 989"},
    { "_id": 14, "name": "Viola", "address": "Sideway 1633"}
]

x = mycol.insert_many(mylist)
```

- a. `mycol.find({}, {"name": 1, "address": 0 })`:
 - b. `mycol.find({}, {"_id": 1, "name": 1, "address": 1 })`:
 - c. `mycol.find({}, {"_id": 1, "name": 1, "address": 0 })`:
 - d. `mycol.find({}, {"_id": 0, "name": 1, "address": 1 })`:
20. ¿El siguiente código qué acción realiza?


```
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]

mycol.drop()
```

- Borra los campos de la colección que se llama “drop”.
- Borra toda la colección “customers”.
- Borra las columnas de la colección “drop”.
- Borra de la colección “customer” los campos que coincidan con la palabra “drop”.

Respuestas

- a
- b
- c
- b
- d
- b
- b
- d
- a
- d
- b
- b
- b
- b
- b
- b
- c
- d
- b

Lista de cotejo: Funcionalidad de proyecto

Para evaluar la funcionalidad del proyecto la tallerista cotejará contra la siguiente lista:

Funcionalidad	SI	NO	Comentarios
Componentes del <i>frontend</i>			

Está presente Bootstrap en cada página generada.			
Existen las plantilla en Jinja para el encabezado y pies.			
Existe un menú que lleva al listado de registros de los 3 módulos.			
Existe un menú para <i>login</i> y salir.			
Existen 3 formularios para agregar y actualizar los módulos de Libros, Usuarías y Préstamo.			
Existe una página de contacto.			
Componentes del backend			
Está ejecutándose el proyecto en flask.			
Existe un módulo de Libros con 4 métodos: agregar, listar, actualizar y eliminar.			
Existe un módulo de Usuarías con 4 métodos: agregar, listar, actualizar y eliminar.			
Existe un módulo de Préstamos con 4 métodos: agregar, listar, actualizar y eliminar.			
Existe un módulo para Login y Salir.			
Componentes de la base de datos			
Existe una colección de Libros con los campos id, Título, autor, prólogo y número de páginas.			
Existe una colección de Usuarías con los campos id, Nombre, apellido y correo electrónico.			
Existe una colección de Préstamos con los campos id, id_libro, id_usuario, fecha_de_prestamo.			
Existe una colección de User con los campos id, email y password.			

Taller 3: Introducción a ciencia de datos con Python

Cuestionario: Conocimiento de conceptos y herramientas

Consistirá en 20 reactivos de opción múltiple en los siguientes temas:

1. Ambiente virtual en Python.
2. Notebooks de Python.
3. Pandas de Python.
4. Manejo de datos.
5. Visualización de datos.

Reactivos

1. ¿Qué problema soluciona los ambientes virtuales de Python?
 - a. Para crear un subdirectorio del proyecto.
 - b. Incluir versiones de bibliotecas para diferentes programas.
 - c. Para mejorar el uso de las variables.
 - d. Para ejecutar un sistema operativo sobre otro.
2. ¿Cuál es la ventaja de usar Jupyter Notebooks?
 - a. Ejecutar instrucción a instrucción para poder hacer cambios dinámicamente sin tener que ejecutar el programa completo.
 - b. Reducir el consumo de memoria RAM.
 - c. Acelerar el proceso.
 - d. No utilizar una base de datos.
3. ¿En qué formato de archivo los datos están ordenados en renglones y para separar las columnas se utilizan comas?
 - a. Json.
 - b. Excel.
 - c. CSV.
 - d. Txt.
4. En el lenguaje Markdown, ¿Qué se obtiene con la siguiente línea de código?
`[A](B "C")`
 - a. Un enlace con el texto A a la URL B y con título C.
 - b. Un enlace con la URL A con un texto B y un argumento C.
 - c. Pone en negritas (*bold*) el texto A y le da formato de centrado.
 - d. Multiplica A por B y C es el número de decimales.
5. ¿Cuál es la diferencia entre una celda de código y una celda Markdown en un Jupyter Notebook?

- a. En la celda de código se ejecuta código Python y solo puede haber una instrucción. Y la celda Markdown imprime texto cuando se ejecuta.
 - b. La celda Markdown imprime texto y la celda de código ejecuta código de Python.
 - c. La celda Markdown imprime texto cuando se ejecuta y solo puede ejecutar una línea. En la celda de código se ejecuta código Python.
 - d. En ambas se puede ejecutar código Python o Markdown, la diferencia es que las celdas Markdown tiene en su mayoría texto formateado y en las celdas Python tiene en su mayoría código de Python
6. En el siguiente ejemplo, la función raíz es declarada en la celda 1. ¿Es necesario volver a declarar la función “raíz” en la celda 2?

```
[1]: import numpy as np
      def raiz(x):
          return x * x

[2]: x = np.random.randint(1, 10)
      y = raiz(x)
      print('La raíz cuadrada de %d es %d' % (x, y))
```

- a. No, porque el código se ejecuta desde el principio cada vez que se hace un cambio a cualquier celda.
 - b. Sí, porque se pierden los datos anteriores cuando se ejecuta cada celda.
 - c. No, porque el código es persistente en el sistema hasta que se reinicie el Notebook.
 - d. Sí, porque no se habilitó la opción de persistencia.
7. ¿Cuál es la diferencia entre datos series y *dataframes* en Pandas?
- a. Los *dataframes* son un subconjunto de los datos series.
 - b. Los *dataframes* son una colección de datos series.
 - c. En los Datos series cada columna tienen un índice y en los *dataframes* no.
 - d. Los primeros son para números y los segundos para números y cadenas.
8. ¿Qué devuelve la función de Pandas DataFrame.head(n=5)?
- a. El nombre de las columnas de un *dataframe*.
 - b. Muestra los primeros 4 renglones del dataframe.
 - c. Muestra los primeros 5 renglones del dataframe.
 - d. Muestra la cabecera del archivo.
9. ¿Cuál es la salida del siguiente código?

```
df = pd.DataFrame({'Peso':[45, 88, 56, 15, 71], 'Name':['Juan, 'Rosario', 'Miguel', 'Abel', 'Ruben'], Edad:[14, 25, 55, 8, 21]})
```

```
df.dtypes
```

- a. Peso int32
- Nombre object

Edad int32

dtype: object

b. Peso float

Nombre object

Edad float

dtype: object

c. Peso int64

Nombre object

Edad int64

dtype: object

d. Peso int64

Nombre int64

Edad int64

dtype: object

10. ¿Cuál es la diferencia entre la función loc e iloc de Pandas?

- a. loc selecciona datos por el la etiqueta de la columna e iloc por el índice de la columnas.
- b. loc sirve para hacer búsquedas de un dato por medio de un filtro e iloc muestra todos los datos.
- c. Ambas sirven para lo mismo, la diferencia es que iloc es una versión mejorada de loc.
- d. loc aloja memoria para un variable e iloc libera el espacio de la memoria.

11. ¿Cuál es la función para calcular el promedio de una columna en Pandas?

- a. df.mean()
- b. df.avg()
- c. No existe, así queda la implementación df.sum()/df.len()
- d. df.promedio()

12. ¿Cuál es la salida del siguiente código?

```
df = pd.DataFrame({'n_piernas': [2, 4, 4, 6], 'n_alas': [2, 0, 0, 0]}, index=['halcón', 'perro', 'gato', 'hormiga'])
```

```
df.value_counts()
```

- a. n_piernas n_alas

4	0	2
6	0	1
2	2	1

b. n_piernas n_alas

4	0	2
6	0	2

c. n_piernas n_alas

4	0	1
6	0	2
2	2	1

d. n_piernas n_alas

6	0	1
2	2	1

13. ¿Cuál es la función correcta para graficar con Matplotlib un histograma del conjunto de datos x con n número de clases?

- a. `plt.hist(x, n)`
- b. `plt.hist(bins=n, x)`
- c. `x=plt.hist(n)`
- d. `plt.hist(x, bins=n)`

14. ¿Cuál es la salida del siguiente código?

```
df = pd.DataFrame({'year': [2015, 2016],
                   'month': [2, 3],
                   'day': [4, 5]})
```

- a. 0 2015-04-02
1 2016-04-02
- b. 0 04-02-2015
1 05-03-2016
- c. 0 2016-03-05
1 2015-02-04

d. 0 2015-02-04

1 2016-03-05

15. ¿Qué arroja el siguiente código?

```
ser = pd.Series([390., 350., 30., 20.], index=['Halcón', 'Halcón', 'Perico', 'Perico'], name="Max Speed")
```

```
ser.ser.groupby(["a", "b", "a", "b"]).mean()
```

a. a 185.0

b 210.0

b. a 210.0

b 185.0

c. a 370.0

b 25.0

d. a 25.0

b 370.0

16. Del siguiente código ¿cuál es la salida?

```
np.prod([[1.,2.],[3.,4.]])
```

a. 24.0

b. $[3],[12]$

c. $[1., 2.], [3., 4.]$

d. Error

17. Del siguiente código ¿cuál es la salida?

```
df = pd.DataFrame({"nombre": ['Alfredo', 'Bartolome', Carla],
```

"apellido": [np.nan, 'Pérez', 'Estrada'],

```
"fecha_nacimiento": [pd.NaT, pd.Timestamp("1940-04-25")],
```

```
pd.NaT]})
```

df.dropna()

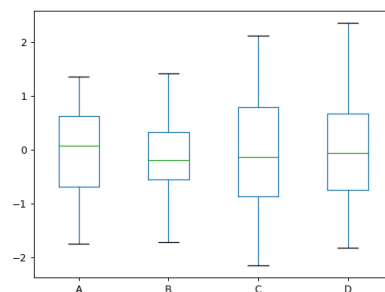
- a. 1 Alfredo np.nan pd.NaT
- b. 1 Bartolome Pérez 1940-05-25
- c. 1 Carla Estrada pd.NaT
- d. 1 Bartolome Pérez 25-05-1940

18. Del siguiente código ¿cuál es la salida?

```
x = pd.Index([10, 100, 1, 1000])
x.sort_values()
```

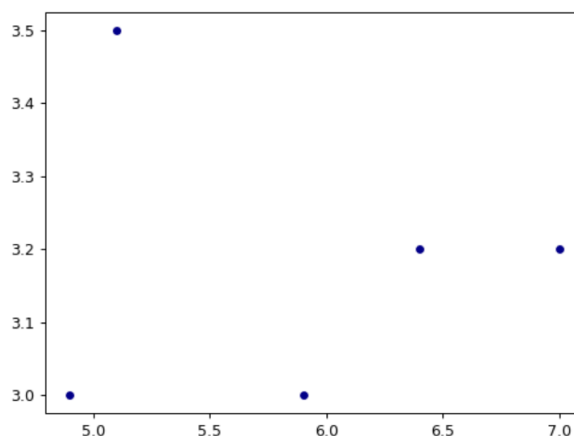
- a. Int64Index([1, 10, 100, 1000], dtype='int64')
- b. Int64Index([1000, 100, 10, 1], dtype='int64')
- c. Int64Index([10, 100, 1, 1000], dtype='int64')
- d. Int64Index([0, 1, 10, 100], dtype='int64')

19. De la siguiente gráfica, ¿cuál fue el código que la generó?



- a. grafica = df.plot.box()
- b. grafica = df.plot.hexbin()
- c. grafica = df.plot.hist()
- d. grafica = df.plot.kde()

20. La siguiente gráfica se generó con la función df.plot.scatter(), ¿cuál es el contenido de df?



- a. pd.DataFrame([[3.5, 5.1, 0], [3.0, 4.9, 0], [3.2, 7.0, 1], [3.2, 6.4, 1], [3.0, 5.9, 2]], columns=['a', 'b', 'c'])
- b. pd.DataFrame([[5.1, 3.5, 0], [4.9, 3.0, 0], [7.0, 3.2, 1], [6.4, 3.2, 1], [5.9, 3.0, 2]],


```
columns=['a', 'b', 'c'])
```

- c. `pd.DataFrame([[5.1, 3.5, 0], [4.9, 3.0, 0], [7.0, 3.2, 1], [6.4, 3.2, 1], [5.9, 3.0, 2], [4, 3.0, 2]], columns=['a', 'b', 'c'])`
- d. `pd.DataFrame([[4.9, 3.0, 0], [7.0, 3.2, 1], [6.4, 3.2, 1], [5.9, 3.0, 2]], columns=['a', 'b', 'c'])`

Respuestas

1. b
2. a
3. c
4. a
5. b
6. c
7. b
8. c
9. c
10. a
11. a
12. a
13. d
14. a
15. b
16. a
17. b
18. a
19. a
20. b

Lista de cotejo: Funcionalidad de proyecto

Funcionalidad	SI	NO	Comentarios
Análisis básicos			
Se cuantifican variables de forma independiente.			

Se obtienen estadísticas básicas de variables independientes.			
Se visualizan las variables de forma independiente.			
Se normalizan las variables en caso necesario.			
Análisis condicionados			
Se identifican variables condicionadas.			
Se obtienen estadísticas básicas de casos de variables condicionados.			
Se visualizan los casos de las variables condicionadas.			
Se normalizan los casos de las variables condicionadas si es necesario.			
En caso de estar presentes se analizan casos en términos de temporalizados.			
Interpretación			
Se identifican casos interesantes.			
Se proporciona evidencia sobre los casos interesantes.			
Se alcanza alguna conclusión sobre los datos.			



Escuela de Código para PILARES Tabla de especificaciones e Instrumentos de evaluación - MP por Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México se distribuye bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).