

PYTHON PARA OPTIMIZACIÓN DE PROCESOS DE ADMISIÓN Y TAREAS DE GESTIÓN DE SOLICITUDES EN RESIDENCIA DE ESTUDIANTES

1. Resumen

Trabajo en una residencia de estudiantes, cubriendo el puesto de Recepcionista/Front Desk.

El funcionamiento del alojamiento es similar a Hoteles y Hostels en cuanto a la gestión de reservas, disponibilidad, ocupación, departamentos de limpieza y mantenimiento.

Lo que los diferencia es que para hospedarse en Onix, es requisito formar parte del ámbito académico. Se alojan tanto estudiantes, como profesores, como investigadores.

Hay distintos tipos de estancia: de día (entre una y seis noches), semanal (siete noches), mes (entre uno y tres meses) y estancias de curso (más de 3 meses).

Dependiendo el tipo de estancia será el procedimiento a seguir para formalizar la reserva.

2. Problema

Todas las solicitudes llegan a través del correo, es decir, el primer contacto con los interesados consiste en que completen el primer paso, que es rellenar un formulario en la página web.

(web.https://www.residenciaonix.com/formulario)

El formulario expone los siguientes datos de los solicitantes:

- **Nombre**
- **Primer Apellido**
- **Segundo Apellido**
- **Fecha de nacimiento**
- **Género**
- **Nro de DNI o Pasaporte**
- **Dirección**
- **País**
- **Código Postal**
- **Provincia**
- **Localidad**
- **Teléfono**
- **Teléfono para urgencias**
- **Correo electrónico**
- **Mes de llegada**
- **Mes de salida**
- **Curso**
- **Estudios que realizarás**
- **Universidad o centro**

Una vez lo envían, nos llega a la misma dirección de correo que se utiliza para cualquier consulta de frontdesk, lo cual deriva a que se pierdan entre Spams, Ofertas, consultas de información general, pedidos, albaranes, archivos que nos envían para imprimir, etc. Una posible solución a esto sería que los formularios lleguen a otra dirección de correo electrónico creada específicamente para las solicitudes.

En cuanto los recibimos, los imprimimos y separamos en carpetas físicas de acuerdo a lo que hayan completado en el campo “**Curso**” del formulario.

Separamos los formularios de los solicitantes en dos categorías:

- **Curso 2024/2025 (del 01/09/2024 al 30/06/2025)**
- **Curso 2025/2026 (del 01/09/2025 al 30/06/2026)**

Cargamos los datos de los Formularios en Excel

- **Uno con las solicitudes del Curso 2024/2025**
- **Otro con las solicitudes del Curso 2025/2026**

En este momento del año estamos trabajando en las solicitudes y admisiones del próximo curso. Respondemos a cada uno de los formularios recibidos, escribimos correo por correo lo cual implica mucha pérdida de tiempo.

Sería óptimo trabajar con datos limpios desde el primer momento. Algunos patrones repetitivos identificados:

- Cuando completan en el el Formulario el campo “**Dirección**”, no coincide el formato. Algunos ponen solo el nombre de la ciudad, otros calle, numeración, manzana, código postal, otros ponen una referencia (*Ej. al lado de Sagrada Familia*). Cambiaría el Formulario de tal modo que la dirección escrita deba ser tal como aparece en Google Maps, es decir, una dirección válida.
- En los campos “**Fecha de Entrada**” y “**Fecha de salida**” lo ideal sería definir el formato en el mismo formulario para evitar que utilicen algunos str (ej. “JUNIO”, “verano”, otros utilizan este formato: 01-02-25, otros: 01/02/2025, otros en formato estadounidense, lo cual además genera confusiones sobre todo en meses como Enero y febrero que si lo han puesto en formato estadounidense indicaría otra fecha que también podría ser válida).
- En “**Teléfono**” y “**Teléfono de emergencia**” Aplicaría una opción de prefijo que sea seleccionable y luego que escriban el número, para que quede en el mismo formato.

Al no ser claro el formulario desde el primer momento, en muchos casos tenemos que contactar a los emisores para aclarar dudas, genera ineficiencias en el proceso de recolección de datos, ya que su interpretación a menudo requiere tiempo adicional.

3.Objetivos

- El objetivo general

El objetivo principal de este proyecto es utilizar Python como herramienta para optimizar y automatizar los procesos de gestión de solicitudes y admisión para reducir la carga de trabajo manual y mejorar la eficiencia operativa del departamento de Front Desk en Residencia Onix.

- Objetivos específicos

- Identificar tareas manuales y repetitivas del proceso de admisión que son susceptibles de automatización.
- Implementar soluciones basadas en Python para integrar la información procesada con los sistemas de gestión existentes, reduciendo la necesidad de transcripción manual.
- Reducir tiempos promedios de procesamiento de cada solicitud.
- Disminuir la tasa de errores relacionados a los campos del Formulario.
- Generar informes y dashboards.

4. Aplicación en el procesamiento de solicitudes del próximo curso

Los datos de los solicitantes están cargados en Excel ("Formularios2025-2026.txt").

Para garantizar la privacidad y cumplir con las normativas de protección de datos, los datos empleados son ficticios, no corresponden a información real.

Desde Jupyter importé la librería de pandas

```
!pip install pandas
```

```
import pandas as pd
```

```
import pandas as pd
```

```
df = pd.read_csv('Formularios2025-2026.csv')
```

Si quiero obtener los nombres de los solicitantes:

```
print(df['Nombre'])
```

```
0      Chloe
1       John
2      Raúl
3     Josefa
4       Luis
5      Kenya
6      Sofia
7      Diego
8       Anya
9       Whei
10    Giuseppe
11        Ali
12     Johan
13        Ben
```

```
Name: Nombre, dtype: object
```

Si quiero obtener los datos de Raúl Sosa y John Perez puedo hacerlo rápidamente mediante:

```
print(df.iloc[2]) # Access the third row by position
print(df.loc[1]) # Access the second row by label
```

Nombre	Raúl
Apellido	Sosa
Fecha de nacimiento	11/3/1998
Género	Masculino
DNI/Pasaporte	AD4554215
Dirección	Cortázar
País	Portugal
Código Postal	1221
Provincia	Lisboa
Localidad	Lisboa
Teléfono	478514125
Teléfono de Urgencias	4785542125
Correo Electrónico	raul.sosa@ejemplo.com
Mes de Llegada	SEPTIEMBRE
Mes de Salida	AGOSTO
Estudios que realizarás	Licenciatura en Psicología
Universidad o centro	UB
Curso	2025/2026

Name: 2, dtype: object

Nombre	John
Apellido	Perez
Fecha de nacimiento	8/1/1995
Género	Masculino
DNI/Pasaporte	21456987D
Dirección	Fleming
País	Argentina
Código Postal	5501
Provincia	Mendoza
Localidad	Godoy Cruz
Teléfono	154718085
Teléfono de Urgencias	156407511
Correo Electrónico	lopezjohn@ejemplo.com
Mes de Llegada	OCTUBRE
Mes de Salida	JUNIO
Estudios que realizarás	Grado en Sonología
Universidad o centro	UB
Curso	2025/2026

Si quiero los de Luis Suarez y Josefa García

```
print(df.iloc[4])
print(df.loc[3])
```

Nombre	Luis
Apellido	Suarez
Fecha de nacimiento	15/3/2000
Género	Masculino
DNI/Pasaporte	Y7890123L
Dirección	Provenza 123

País	España
Código Postal	8036
Provincia	Barcelona
Localidad	Barcelona
Teléfono	601234568
Teléfono de Urgencias	601234567
Correo Electrónico	luis.suarez@ejemplo.com
Mes de Llegada	SEPTIEMBRE
Mes de Salida	JUNIO
Estudios que realizarás	Grado en Ingeniería Informática
Universidad o centro	UPC
Curso	2025/2026
Name: 4, dtype: object	
Nombre	Josefa
Apellido	García
Fecha de nacimiento	2/8/1996
Género	Femenino
DNI/Pasaporte	ER4885451
Dirección	Plaza España
País	España
Código Postal	75510
Provincia	Almería
Localidad	Almería
Teléfono	689541225
Teléfono de Urgencias	687456123
Correo Electrónico	josefa.dominguez@ejemplo.com
Mes de Llegada	SEPTIEMBRE
Mes de Salida	OCTUBRE
Estudios que realizarás	Máster Arquitectura paisajística
Universidad o centro	ETSAB
Curso	2025/2026
Name: 3, dtype: object	

Si quiero obtener los valores únicos de ID

```
unique_ID = df['DNI/Pasaporte'].unique()
```

```
print(unique_ID)
```

```
['FR112233445' '21456987D' 'AD4554215' 'ER4885451' 'Y7890123L'
 'JP123456789' 'X123456789M' 'MX54565456' 'IN334455667' 'CN556677889'
 'IT5689562455' 'EG123456789' 'SE667788991' 'AU223344553']
```

[25]:

Si quiero saber cuántas solicitudes hay, lo cual informamos una vez a la semana:

```
len(unique_ID)
```

```
14
```

Para crearle un usuario dentro de Ulysses Cloud necesito sólo algunos de esos datos.

Quiero extraer solo Nombre, Apellido, DNI/Pasaporte, País y asignarlo a la variable huésped

```
huésped = df[['Nombre','Apellido','DNI/Pasaporte', 'País']]
```

```
huésped
```



Nombre	Apellido	DNI/Pasaporte	País
0	Chloe	Dubois	FR112233445 Francia
1	John	Perez	21456987D Argentina
2	Raúl	Sosa	AD4554215 Portugal
3	Josefa	García	ER4885451 España
4	Luis	Suarez	Y7890123L España
5	Kenya	Tanaka	JP123456789 España
6	Sofia	Garcia	X123456789M España
7	Diego	Morales	MX54565456 México
8	Anya	Sharma	IN334455667 India
9	Whei	Lin	CN556677889 China
10	Giusepp e	Lombardi	IT568956245 Italia
11	Ali	Hassan	EG123456789 Egipto
12	Johan	Svensson	SE667788991 Suecia
13	Ben	Carter	AU223344553 Australia

Con lo cual ya puedo generarles el usuario a los huéspedes en el sistema (aún no tienen una reserva asignada, ya que son solicitantes por el momento no sabemos si habrá disponibilidad o no).

Para analizar datos sobre qué países predominan, si hay más huéspedes femeninos que masculinos, y a qué universidad o centro asistirán.

Nombre	Apellido	País	Género	Universidad o centro	
0	Chloe	Dubois	Francia	Femenino	ESADE
1	John	Perez	Argentina	Masculino	UB
2	Raúl	Sosa	Portugal	Masculino	UB
3	Josefa	García	España	Femenino	ETSA B
4	Luis	Suarez	España	Masculino	UPC
5	Kenya	Tanaka	España	Femenino	ESADE
6	Sofía	Garcia	España	Femenino	URL
7	Diego	Morales	México	Masculino	UB
8	Anya	Sharma	India	Femenino	IET
9	Whei	Lin	China	Masculino	UPC
10	Giuseppe	Lombardi	Italia	Masculino	CETT
11	Ali	Hassan	Egipto	Masculino	UB
12	Johan	Svensson	Suecia	Masculino	UPC
13	Ben	Carter	Australia	Masculino	ESADE

El segundo paso del proceso de pre-registro es crear la reserva, para ello necesito sólo algunos datos de los solicitantes.

```
reserva = df[['Nombre', 'Apellido', 'País', 'Género', 'Mes de Llegada', 'Mes de Salida']]
reserva
```

	Nombre	Apellido	País	Género	Mes de Llegada	Mes de Salida
0	Chloe	Dubois	Francia	Femenino	SEPTIEMBRE	AGOSTO
1	John	Perez	Argentina	Masculino	OCTUBRE	JUNIO
2	Raúl	Sosa	Portugal	Masculino	SEPTIEMBRE	AGOSTO
3	Josefa	García	España	Femenino	SEPTIEMBRE	OCTUBRE
4	Luis	Suarez	España	Masculino	SEPTIEMBRE	JUNIO
5	Kenya	Tanaka	España	Femenino	OCTUBRE	JULIO
6	Sofia	Garcia	España	Femenino	SEPTIEMBRE	JUNIO
7	Diego	Morales	México	Masculino	SEPTIEMBRE	AGOSTO
8	Anya	Sharma	India	Femenino	OCTUBRE	FEBRERO
9	Whei	Lin	China	Masculino	SEPTIEMBRE	AGOSTO
10	Giusep e	Lombard i	Italia	Masculino	SEPTIEMBRE	JULIO
11	Ali	Hassan	Egipto	Masculino	SEPTIEMBRE	AGOSTO
12	Johan	Svensso n	Suecia	Masculino	SEPTIEMBRE	JUNIO

13 Ben Carter Australia Masculino

SEPTIEMBR
E AGOST
O

Supongamos que sólo quiero Nombre, Apellido y Fecha de nacimiento:
df.iloc[0:14, 0:3]

	Nombre	Apellido	Fecha de nacimiento
0	Chloe	Dubois	29/7/1999
1	John	Perez	8/1/1995
2	Raúl	Sosa	11/3/1998
3	Josefa	García	2/8/1996
4	Luis	Suarez	15/3/2000
5	Kenya	Tanaka	11/2/2002
6	Sofia	Garcia	22/7/2001
7	Diego	Morales	28/6/1998
8	Anyia	Sharma	3/9/2001
9	Whei	Lin	14/6/1998
10	Giuseppe	Lombardi	11/3/1994
11	Ali	Hassan	25/9/2003

12	Johan	Svensson	31/7/1995
----	-------	----------	-----------

13	Ben	Carter	4/4/1996
----	-----	--------	----------

Hay datos que sólo servirán como datos de contacto, los podemos filtrar:
df.loc[0:14,'Nombre':'Teléfono de Urgencias']

Necesito sólo Nombre, Apellido y Número de teléfono para agendarlos en WhatsApp.
contacto = df[['Nombre','Apellido', 'Teléfono']]
contacto

	Nombre	Apellido	Teléfono
0	Chloe	Dubois	671134628
1	John	Perez	154718085
2	Raúl	Sosa	478514125
3	Josefa	García	689541225
4	Luis	Suarez	601234568
5	Kenya	Tanaka	698551454
6	Sofia	Garcia	606789101
7	Diego	Morales	535656865

8	Anya	Sharma	62611223 3
9	Whei	Lin	56696656 6
10	Giuseppe	Lombardi	64455667 8
11	Ali	Hassan	63500112 2
12	Johan	Svensson	63744556 6
13	Ben	Carter	63988900 2

Pueden ver todos mis códigos empleados en

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-826b414eaa32d6037739ed3f76e35c735b213ac6>

Mi notebook:

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-826b414eaa32d6037739ed3f76e35c735b213ac6>