**# Función para configurar el controlador y la página de Urban Routes**

**def setup():  
 driver = webdriver.Chrome()  
 driver.get(data.URBAN\_ROUTES\_URL)  
 urban\_routes\_page = UrbanRoutesPage(driver)  
 return driver, urban\_routes\_page  
  
# Función para realizar una prueba de reserva de taxi  
def test\_booking():  
 driver, urban\_routes\_page = setup()  
 urban\_routes\_page.set\_from\_address(data.ADDRESS\_FROM)  
 urban\_routes\_page.set\_to\_address(data.TO\_ADDRESS)  
 urban\_routes\_page.click\_request\_taxi(data.REQUEST\_TAXI)  
 urban\_routes\_page.select\_category(data.SELECT\_CATEGORY)  
 urban\_routes\_page.enter\_phone\_number(data.PHONE\_NUMBER)  
 urban\_routes\_page.click\_next\_button()  
 assert urban\_routes\_page.is\_phone\_submitted(), "Phone number not submitted"  
 driver.quit()  
  
# Función para realizar una prueba de agregar una tarjeta de crédito  
def test\_add\_credit\_card():  
 driver, urban\_routes\_page = setup()  
 # Agregar una tarjeta de crédito  
 urban\_routes\_page.select\_payment\_method()  
 assert urban\_routes\_page.is\_payment\_method\_added(), "Payment method not added"  
 driver.quit()  
  
# Función para realizar una prueba de pedido adicional  
def test\_additional\_order():  
 driver, urban\_routes\_page = setup()  
 urban\_routes\_page.add\_comment\_for\_driver("Traer los snacks")  
 urban\_routes\_page.activate\_blankets\_and\_tissues()  
 urban\_routes\_page.add\_ice\_creams()  
 urban\_routes\_page.request\_taxi\_again()  
 assert urban\_routes\_page.is\_taxi\_requested(), "Taxi request failed"  
 urban\_routes\_page.wait\_for\_bender()  
 assert urban\_routes\_page.wait\_for\_bender(), "Bender not found"  
 driver.quit()  
  
# Función principal para ejecutar todas las pruebas  
if \_\_name\_\_ == "\_\_main\_\_":  
 test\_booking()  
 test\_add\_credit\_card()  
 test\_additional\_order()**  
 # Configurar la dirección  
 urban\_routes\_page.set\_from\_address("East 2nd Street, 601") # Usar la dirección definida en data.py  
 urban\_routes\_page.set\_to\_address("1300 1st St") # Usar la dirección definida en data.py  
 urban\_routes\_page.click\_request\_taxi()  
 assert urban\_routes\_page.is\_booking\_confirmed(), "Booking was not confirmed"  
  
 # seleccionar la tarifa "Confort"  
 urban\_routes\_page.select\_category("Comfort")  
 assert urban\_routes\_page.is\_category\_selected("Comfort"), "Comfort category not selected"  
  
 # Rellenar el número de teléfono  
 urban\_routes\_page.click\_phone\_field()  
 urban\_routes\_page.enter\_phone\_number("+12312312312")  
 urban\_routes\_page.click\_next\_button()  
 assert urban\_routes\_page.is\_phone\_submitted(), "Phone number not submitted"  
  
 # Obtener y confirmar código SMS  
 urban\_routes\_page.retrieve\_sms\_code()  
 assert sms\_code, "SMS code not retrieved successfully"  
  
 # Agregar una tarjeta de Crédito  
 urban\_routes\_page.select\_payment\_method()  
 assert urban\_routes\_page.is\_payment\_method\_added(), "Payment method not added"  
  
 # Escribir un mensaje para el conductor  
 urban\_routes\_page.add\_comment\_for\_driver("Traer los snacks")  
 assert urban\_routes\_page.is\_comment\_added("Traer los snacks"), "Comment not added"  
  
 # Pedir una manta y pañuelos.  
 urban\_routes\_page.activate\_blankets\_and\_tissues()  
 assert urban\_routes\_page.are\_blankets\_activated(), "Blankets not activated"  
  
 # Pedir 2 helados.  
 urban\_routes\_page.add\_ice\_creams()  
 assert urban\_routes\_page.are\_ice\_creams\_added(), "Ice creams not added"  
  
 # Aparece el modal para buscar un taxi.  
 urban\_routes\_page.request\_taxi\_again()  
 assert urban\_routes\_page.is\_taxi\_requested(), "Taxi request failed"  
  
 # Esperar a que aparezca la información del conductor en el modal  
 urban\_routes\_page.wait\_for\_bender()  
 assert urban\_routes\_page.wait\_for\_bender(), "Bender not found"  
  
 # Esperar y cerrar la página  
 urban\_routes\_page.wait\_and\_close\_page(5)