LINHA DE CÓDIGOS ALEATÓRIO

```
mport sys
import time
import types
import sl4a
try:
  import gdata.docs.service
except ImportError:
  gdata = None

droid = sl4a.Android()


def event_loop():
  for i in range(10):
    time.sleep(1)
    droid.eventClearBuffer()
    time.sleep(1)
    e = droid.eventPoll(1)
    if e.result is not None:
      return True
  return False

def test_imports():
  try:
    import termios
    import bs4 as BeautifulSoup
    import pyxmpp2 as xmpp
    from xml.dom import minidom
  except ImportError:
    return False
  return True

def test_clipboard():
  previous = droid.getClipboard().result
  msg = 'Hello, world!'
  droid.setClipboard(msg)
  echo = droid.getClipboard().result
  droid.setClipboard(previous)
  return echo == msg


def test_gdata():
  if gdata is None:
    return False

  # Create a client class which will make HTTP requests with Google Docs server.
  client = gdata.docs.service.DocsService()

  # Authenticate using your Google Docs email address and password.
  username = droid.dialogGetInput('Username').result
  password = droid.dialogGetPassword('Password', 'For ' + username).result
  try:
    client.ClientLogin(username, password)
  except:
    return False

  # Query the server for an Atom feed containing a list of your documents.
  documents_feed = client.GetDocumentListFeed()
  # Loop through the feed and extract each document entry.
  return bool(list(documents_feed.entry))


def test_gps():
  droid.startLocating()
  try:
    return event_loop()
  finally:
    droid.stopLocating()


def test_battery():
  droid.batteryStartMonitoring()
  time.sleep(1)
  try:
    return bool(droid.batteryGetStatus())
  finally:
    droid.batteryStopMonitoring()

def test_sensors():
  # Accelerometer, once per second.
  droid.startSensingTimed(2, 1000)
  try:
    return event_loop()
  finally:
    droid.stopSensing()


def test_speak():
  result = droid.ttsSpeak('Hello, world!')
  return result.error is None


def test_phone_state():
  droid.startTrackingPhoneState()
  try:
    return event_loop()
  finally:
    droid.stopTrackingPhoneState()


def test_ringer_silent():
  result1 = droid.toggleRingerSilentMode()
  result2 = droid.toggleRingerSilentMode()
  return result1.error is None and result2.error is None


def test_ringer_volume():
  get_result = droid.getRingerVolume()
  if get_result.error is not None:
    return False
  droid.setRingerVolume(0)
  set_result = droid.setRingerVolume(get_result.result)
  if set_result.error is not None:
    return False
  return True


def test_get_last_known_location():
  result = droid.getLastKnownLocation()
  return result.error is None


def test_geocode():
  result = droid.geocode(0.0, 0.0, 1)
  return result.error is None


def test_make_toast():
  result = droid.makeToast('Hello, world!')
  return result.error is None
```

```
def test_vibrate():
  result = droid.vibrate()
  return result.error is None


def test_notify():
  result = droid.notify('Test Title', 'Hello, world!')
  return result.error is None


def test_get_running_packages():
  result = droid.getRunningPackages()
  return result.error is None


def test_alert_dialog():
  title = 'User Interface'
  message = 'Welcome to the SL4A integration test.'
  droid.dialogCreateAlert(title, message)
  droid.dialogSetPositiveButtonText('Continue')
  droid.dialogShow()
  response = droid.dialogGetResponse().result
  return response['which'] == 'positive'


def test_alert_dialog_with_buttons():
  title = 'Alert'
  message = ('This alert box has 3 buttons and '
             'will wait for you to press one.')
  droid.dialogCreateAlert(title, message)
  droid.dialogSetPositiveButtonText('Yes')
  droid.dialogSetNegativeButtonText('No')
  droid.dialogSetNeutralButtonText('Cancel')
  droid.dialogShow()
  response = droid.dialogGetResponse().result
  return response['which'] in ('positive', 'negative', 'neutral')


def test_spinner_progress():
  title = 'Spinner'
  message = 'This is simple spinner progress.'
  droid.dialogCreateSpinnerProgress(title, message)
  droid.dialogShow()
  time.sleep(2)
  droid.dialogDismiss()
  return True


def test_horizontal_progress():
  title = 'Horizontal'
  message = 'This is simple horizontal progress.'
  droid.dialogCreateHorizontalProgress(title, message, 50)
  droid.dialogShow()
  for x in range(0, 50):
    time.sleep(0.1)
    droid.dialogSetCurrentProgress(x)
  droid.dialogDismiss()
  return True


def test_alert_dialog_with_list():
  title = 'Alert'
  droid.dialogCreateAlert(title)
  droid.dialogSetItems(['foo', 'bar', 'baz'])
  droid.dialogShow()
  response = droid.dialogGetResponse().result
  return True


def test_alert_dialog_with_single_choice_list():
  title = 'Alert'
  droid.dialogCreateAlert(title)
  droid.dialogSetSingleChoiceItems(['foo', 'bar', 'baz'])
  droid.dialogSetPositiveButtonText('Yay!')
  droid.dialogShow()
  response = droid.dialogGetResponse().result
  return True


def test_alert_dialog_with_multi_choice_list():
  title = 'Alert'
  droid.dialogCreateAlert(title)
  droid.dialogSetMultiChoiceItems(['foo', 'bar', 'baz'], [])
  droid.dialogSetPositiveButtonText('Yay!')
  droid.dialogShow()
  response = droid.dialogGetResponse().result
  return True

def test_wifi():
  result1 = droid.toggleWifiState()
  result2 = droid.toggleWifiState()
  droid.toggleWifiState(True)  # Make sure wifi ends up ON, as it interferes with other tests
  return result1.error is None and result2.error is None

if __name__ == '__main__':
  for name, value in list(globals().items()):
    if name.startswith('test_') and isinstance(value, types.FunctionType):
      print('Running %s...' % name, end=' ')
      sys.stdout.flush()
      if value():
        print(' PASS')
      else:
        print(' FAIL')
```